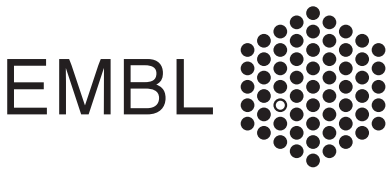


This dissertation is submitted for the degree of Doctor of Philosophy

Community Classification of the Protein Universe

Matthew Jacob Jeffryes

September 2018

Corpus Christi College,
University of Cambridge

EMBL-EBI

Typeset in 11pt Minion Pro with one-and-a-half spacing by XeLaTeX 0.99998.

To Charis

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text.

It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution.

It does not exceed the word limit of 60,000 words defined by the Biology Degree Committee.

Matthew Jeffryes
September 2018

Acknowledgements

I am most thankful to my supervisor Alex Bateman. Alex has always made time for discussions with me, and his guidance and encouragement has kept me on track through the course of my studies.

I thank the members of my thesis advisory committee: Toby Gibson, Nick Goldman, and Pietro Lió for their valuable advice and encouragement. I also thank Maria Liakata for many useful discussions early on in my project, particularly regarding natural language processing.

I thank Rob Finn and the sequence family team at EMBL-EBI. In particular, Aurélien Luciani for his help with the HMMER web server, Matthias Blum for developing Pronto and assisting me in extending it, InterPro curators Lorna Richardson and Amaia Sangrador for answering my questions on curation and giving me feedback, and Pfam curator Sara El-Gebali for the same, and for proofreading and her support.

I thank the other members Alex Bateman's 'small but perfectly formed' research group. Ananth Prakash for help in many areas and for setting an exceptional example in meeting thesis submission deadlines, Aleix Lafita for many discussions and for proofreading, Wolfram Höps and Irina Ponamareva for their brief but productive time in the lab, and Lowri Williams for proofreading.

I am endlessly grateful to the European Molecular Biology Laboratory, for funding me and for giving me the opportunity to work alongside so many incredible people. It has been the greatest privilege to be part of EMBL. I thank the other 'predocs' at EBI and throughout EMBL for their friendship and solidarity. I have been humbled by their scientific brilliance and vision.

I thank my parents for nurturing my interest in science, and for unwaveringly supporting me in everything I have done. And my wife Charis, for everything.

Summary

Protein family databases are an important resource for biologists seeking to characterise the function of proteins, the structure of their domains, and their localisation within the cell. Operating a protein family database requires the identification of families, and the curation of literature related to the family. This labour is currently performed by skilled professional curators, whose abilities are a scarce resource. In this thesis, I have developed methods to enable some of this labour to be performed by the community of protein sequence similarity search users.

In the first chapter, I review the history of protein sequence and protein family databases, and how the abstract concept of a protein family is expressed as a computational model. I review in greater detail the protein family database Pfam, and the software package HMMER, which uses hidden Markov models to search protein sequence databases.

In the second chapter, I explore how the quality of computational models for a protein family can be measured, and how these measurements might be used to assess the quality of community-sourced protein family models. I then investigate how a protein sequence similarity search can be rapidly analysed for overlap with existing protein families in Pfam, using locality sensitive hashing.

In the third chapter, I discuss the use of literature search in protein family database curation, and the existing literature resources used by protein family database curators. I then develop a system for performing literature search based on protein families, exploiting the manually annotated links between literature and proteins found in the Swiss-Prot subset of the UniProt protein database.

In the fourth chapter, I develop a web application for analysing the results of protein sequence similarity searches, using the methods discussed in the second chapter, and for performing literature search based on the results of protein sequence similarity search, using the methods discussed in the third chapter.

In the fifth chapter, I develop a web application which applies the methods developed in the third chapter to the task of curation of the protein classification resource, InterPro.

Contents

Acronyms

tf–idf term frequency–inverse document frequency. 65, 66

API application programming interface. 76–78

DUF domain of unknown function. 81, 88

EAV ENTH/ANTH/VHS. 29–34

EMBL European Molecular Biology Laboratory. 10

EMBL-EBI European Bioinformatics Institute. 10, 20, 52, 76, 87

HMM hidden Markov model. 5, 12, 13, 15–20, 28, 31, 33, 44, 48, 51, 88

MeSH Medical Subject Headings. 52, 65–67

NCBI National Centre for Biotechnology Information. 65

NER named entity recognition. 57–60, 62, 89

NLM National Library of Medicine. 52, 53, 57, 62, 80

NLP natural language processing. 20, 55

PIR Protein Information Resource. 10

PMC PubMed Central. 52, 60–62, 64–68, 78

PMCID PubMed Central identifier. 65, 78

PMID PubMed identifier. 65

SIB Swiss Institute of Bioinformatics. 10

URL uniform resource locator. 81

UUID universally unique identifier. 78

XML Extensible Markup Language. 52, 61, 76

List of Figures

List of Tables

1. Introduction

> Databases do not inspire excitement.
>
> *Margaret Oakley Dayhoff, following a failed grant application (Strasser, 2010).*

> After watches and chocolates, [Switzerland] is best known abroad for Swiss-Prot.
>
> *Amos Bairoch reacting to letters of support for his protein database, at the time under the threat of closure (Butler, 1996).*

Since Fred Sanger published the sequence of insulin in a series of papers in the early 1950s, biologists have been faced with an ever increasing volume of protein sequences (Sanger and Tuppy, 1951a,b; Sanger and Thompson, 1953a,b). Having decoded bovine insulin, Sanger turned his attention to the differences in the sequence of the protein between different species, and this work led Crick (1958) to foresee the foundation of a new field concerned with the inference of evolutionary relationships from these differences, which he called 'protein taxonomy'. The concept of a protein family was developed to identify proteins for which an evolutionary relationship has been identified (Dayhoff, W. C. Barker *et al.*, 1974; Dayhoff, 1974). Latterly, protein sequences are inferred from DNA sequence, rather than determined by sequencing the protein directly, as Sanger did. But as sequencing became more routine, it became unfeasible to identify such evolutionary relationships by eye. To construct her *Atlas of Protein Sequence and Structure*, Margaret Oakley Dayhoff (1976) pioneered the use of computational techniques to identify related 'superfamilies' of proteins. The first volume of the *Atlas*, published in 1965, was under 100 pages long, and contained 70 protein sequences. The now standard single letter abbreviations for amino acids were devised by Dayhoff in order to make the representation of sequences more compact in the *Atlas* (see figure 1.1). This system also enabled visual analysis of sequence conservation in the alignments which accompanied the entries for homologous

sequences. Dayhoff is now considered to be one of the founders of bioinformatics, and protein classification is therefore one of the foundational problems for the field (Strasser, 2010; Hagen, 2011).

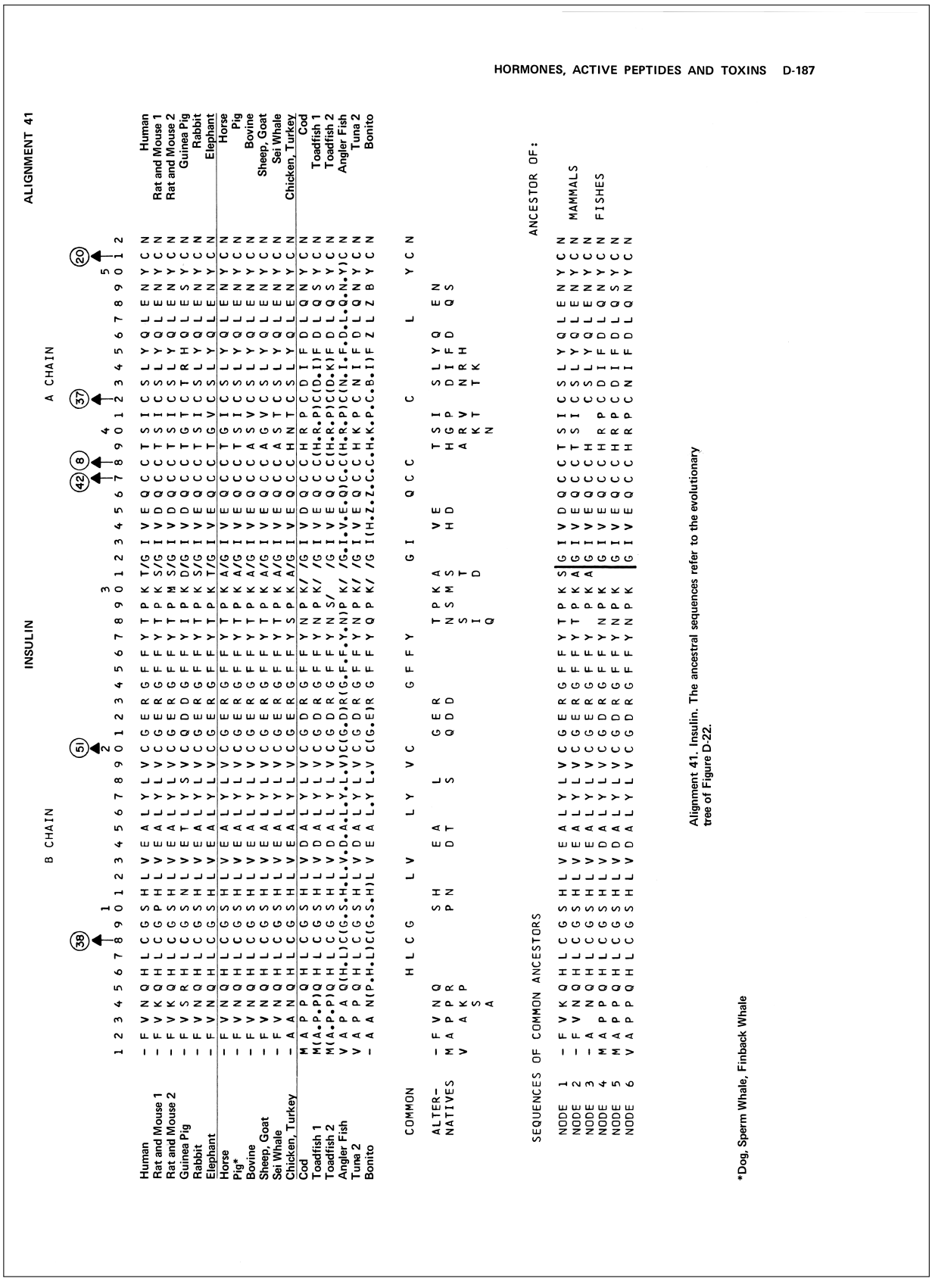
1.1. Sequence Databases

Dayhoff's *Atlas* led directly to the establishment of the Protein Identification Resource in 1984, which made protein sequence data available digitally via pre-Internet networks and magnetic tape. The Protein Identification Resource later became the Protein Information Resource and then the international collaboration PIR-International (W. Barker *et al.*, 1998; George *et al.*, 1986; Sidman *et al.*, 1988). In 1986, the Swiss-Prot protein database was first released by Amos Bairoch (Bairoch and Boeckmann, 1991). Swiss-Prot was later maintained jointly by Bairoch's group at the University of Geneva, and the European Molecular Biology Laboratory (EMBL) Data Library group. Bairoch went on to be a founder of the Swiss Institute of Bioinformatics (SIB), which was founded partially to provide a funding vehicle for the database. The EMBL group evolved into the European Bioinformatics Institute (EMBL-EBI), which was founded as a custodian for Swiss-Prot and other emerging bioinformatics resources. To complement Swiss-Prot, the TrEMBL database was created in 1996. In contrast to Protein Information Resource (PIR) and Swiss-Prot, the entries in TrEMBL are automatically generated from translated nucleotide sequences (Apweiler *et al.*, 1996). After weathering funding crises, which resulted in Swiss-Prot temporarily charging commercial users, all three of these resources were integrated in the UniProt consortium from 2002 (Abbott, 1998; Butler, 2002).

UniProt now provides a single, freely available, international resource for protein information. The UniProt Knowledgebase integrates 550,000 sequences from Swiss-Prot, which are manually annotated by staff, and 107 million sequences from TrEMBL, which are automatically annotated (The UniProt Consortium, 2017).

1.2. Protein Classification

Protein classification has also evolved since the *Atlas*. Today, protein family databases are a vital component of the bioinformatics toolbox. We define protein family databases as those which aim to provide insight about homology given a query protein sequence. We can visualise protein sequences as existing on a vast multidimensional 'protein space'. Closely related sequences are neighbours, while those which diverged billions of years ago will be more distant from each other. This space can be partitioned such that sequences are grouped with the other members



Alignment 41. Insulin. The ancestral sequences refer to the evolutionary tree of Figure D-22.

Figure 1.1. The alignment of Insulin from the *Atlas of Protein Sequence and Structure* (Dayhoff, 1972). This demonstrates how Dayhoff's single letter abbreviations for amino acids allowed for visual analysis of alignments. The longest alignments in the *Atlas* were printed on fold-dout sheets, up to 80cm long.

of their protein family, illustrated in figure 1.2. In order to draw these boundaries and assign sequences to families, we must infer their homology. This is the problem inherent in building a protein family database.

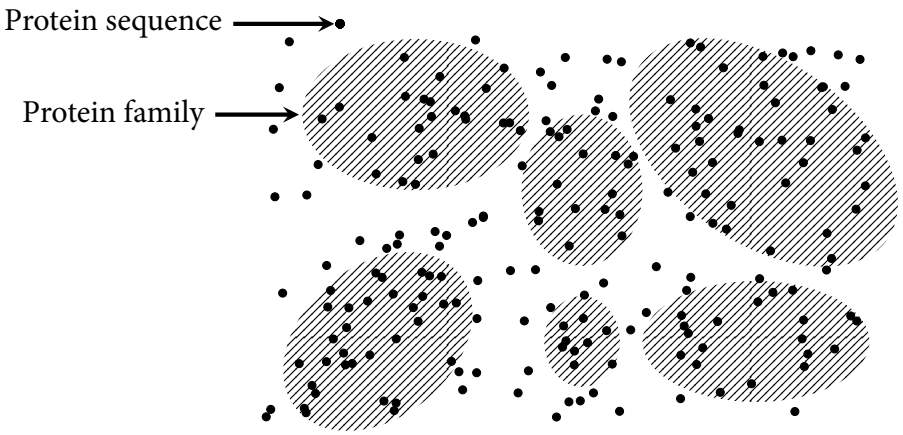


Figure 1.2. An idealised view of families in sequence space. No sequence is contained in more than one family.

The term 'protein family' has been used differently by different authors. Dayhoff referred to only very closely related groups of proteins, which had been identified manually and for which sequence conservation was obvious, as protein families. She referred to groupings of more distantly related proteins, identified by statistical and computational methods, as superfamilies (Dayhoff, 1976). In contrast, the Pfam database refers to groupings of related proteins (all identified by computational methods) as a protein family, and uses a higher level grouping of 'clan' for more distant relationships, or when it is not possible to create a model which matches the entire group of proteins (Finn, Mistry *et al.*, 2006). Pfam entries are further classified as one of six types: Family, domain, motif, repeat, coiled coil and disordered (Finn, Coggill *et al.*, 2015). In this work, I will use protein family to refer to any grouping of related proteins.

One approach to protein classification is to group proteins based on the genes which code for them. The process by which two proteins become differentiated from their common ancestor and from each other is a series of mutations of their coding sequence. Once an initial duplication event has resulted in two copies of the same gene, the copies are free to evolve independently. Over time, they can accumulate different mutations. Computational phylogeny allows the inference of a family tree for the descendants of a particular gene. In this gene family centric model, a protein family is conceptualised as the translational products of a gene family.

[1] Arguably, an intrinsically disordered protein may have no domains, depending on the definition of 'domain' used. The traditional view, that function was a consequence of a fixed 3D structure is obsolete, and the disordered regions of many proteins are now known to be functional (van der Lee *et al.*, 2014). Regardless of semantics, there is still selective pressure to maintain the sequence of disordered but functional regions.

Alternatively, the classification of proteins can be based upon their domains. A protein is made up of one or more domains.[1] The sequence region which codes for a domain is conserved across different proteins and species, since the domain's function is dependent on the sequence. The domains of a multidomain protein can be members of different families. In this domain-centric model, a protein family is conceptualised as a grouping of protein regions containing similar domains.

To understand the difference between these models we can consider a protein composed of two domains. Under the gene family centric model, we would place the protein in a phylogenetic tree by identifying which of its relatives it is most closely related to. To do this, the protein sequence is considered as a whole without regard to the boundaries of the domains. Under the domain-centric model, each domain is considered in isolation. The regions containing each of the two domains are grouped with regions with which the domains are homologous. The gene family centric model is more concerned with the evolutionary history of the protein, whereas the domain centric model is more concerned with the function of the protein.

1.2.1. Protein Family Databases

Protein family databases generally describe a particular family using a sequence profile, often in the form of a hidden Markov model (HMM) (Eddy, 1998). The profile HMM is a representation of the multiple sequence alignment of a number of representatives of a family. The likelihood that a given sequence is a member of a family (that is, it has homology with all the other members of the family) is thus estimated by the probability of its alignment to this profile HMM. The representative sequences in the alignment may be selected automatically or by a curator depending upon the database.

A protein family database should cover as much of sequence space as possible, while avoiding overlapping sequence profiles.[2] An overlap occurs when a particular region in a protein sequence is a significant match for more than one sequence profile. In this case there are two possibilities. Either the region of the protein sequence in which the overlapping matches lie is a false positive for one or both of sequence profiles, or the sequence profiles match proteins which are homologous. Maximising coverage of sequence space increases the chance of overlap. Each sequence profile added to increase coverage may overlap with the existing profile HMMs in the database. Such overlaps are a result of the fact that HMM sequence profiles are an imperfect model of the underlying homology of the families that they represent.

[2] Some databases have multiple levels of classification. 'Superfamilies' and 'clans' are examples of levels under different classification schemes. It is not a contradiction for a residue to be assigned to different families at different levels, just as humans are placed in both the primates and mammalia.

In some curated databases, groupings of proteins are required to adhere to a consistent taxonomical hierarchy, to encode the fact that each sequence region has a single evolutionary origin. Curators use their research skill and personal expertise to avoid contradictions. For an automated database, this is much more difficult. The EVEREST database simply included overlapping profiles and when queried, presented the potentially contradictory information to the user, requiring them to make a judgement about the true homology based on their biological knowledge (Portugaly *et al.*, 2007).

A protein family database may perform a number of functions. This could include returning the name of the family to which a protein belongs, the region of the protein sequence which is homologous with the rest of the family and possibly additional biological or structural information about the homology. We can broadly categorise protein classification databases, past and present, as either *(i)* curated; or *(ii)* automated. In a curated database, curators determine which classifications are included in the database, whereas an automated database uses an automatic process to generate these classifications. Curated databases are more reliable, but cover a smaller area of protein space, whereas an automated database will cover more of protein space but with less reliable annotations. Additionally, to incorporate new information, an automated database may have to be rebuilt from scratch, making tracking annotations across releases difficult, whereas a curated database may be able to ensure more continuity between releases (Servant *et al.*, 2002). A researcher attempting to classify a newly sequenced protein may first consult a curated database, and will resort to an automated database if the curated database does not cover their new sequence.

Some examples of the first kind of database are Pfam, PRINTS and SUPERFAMILY (Finn, Bateman *et al.*, 2014; Attwood, Croning *et al.*, 2000; Wilson *et al.*, 2007). Some examples of the second kind of database are Pfam-B, EVEREST and SYSTERS (Bateman, Coin *et al.*, 2004; Portugaly *et al.*, 2007; Meinel *et al.*, 2005). A summary of published protein family databases is shown in table 1.2.

All extant databases in table 1.2 are members of the InterPro consortium, which replicates the model of international database collaboration demonstrated by UniProt (Mitchell *et al.*, 2015). The collaboration intended to reduce duplication in the classification of proteins, and enables all databases to be searched simultaneously for annotations of a particular sequence. It is notable that of the automatic databases listed, none are still maintained. In their reasoning for retiring Pfam-B in 2013, Finn, Coggill *et al.* (2015) wrote that the automated database was no longer cost effective to produce: The informative groupings derived by the algorithm largely duplicated curated families, and the novel remainder did not justify the time required to generate the database. The paucity of extant automatic databases, and the decades-long persistence

Database	Introduction	Latest Release	Entries	Annotation	Reference
Prosite	1988	2018-01 2018	1,800	Curated	Sigrist *et al.* (2013)
BLOCKS †	1990	14.3 2007	29,068	Automatic	J. G. Henikoff, Greene *et al.* (2000)
PIR-ALN †	1991	19.0 1998	3,468	Curated	Srinivasarao *et al.* (1999)
PRINTS	1991	42.0 2012	2,156	Curated	Attwood, Coletta *et al.* (2012)
SBASE †	1992	14 2006	8,029	Automatic	Vlahoviček *et al.* (2004)
PRODOM †	1994	2012.1 2015	1,718,157	Automatic	Bru (2005)
Pfam	1996	31.0 2017	16,712	Curated	Finn, Coggill *et al.* (2015)
Pfam-B †	1996	27.0 2013	544,866	Automatic	Finn, Coggill *et al.* (2015)
DOMO †	1997	2001	8,877	Automatic	Gracy and Argos (1998a,b)
SYSTERS †	1997	4 2005	158,153	Automatic	Meinel *et al.* (2005)
SMART	1998	8.0 2016	1,302	Curated	Letunic and Bork (2018)
PANTHER	1998	13.1 2018	15,524	Curated	Mi *et al.* (2015)
TIGRFAMS	2000	15.0 2014	4,488	Curated	Haft *et al.* (2013)
SUPERFAMILY	2001	1.75 2014	15,438	Curated	Oates *et al.* (2015)
HAMAP	2002	2018-01 2018	2,229	Curated	Pedruzzi *et al.* (2015)
PIRSF	2003			Curated	Wu *et al.* (2004)
EVEREST †	2005	2.0 2006	20,029	Automatic	Portugaly *et al.* (2007)

Table 1.2. Protein family databases past and present. This table is based on the review in Bateman and Birney (2000). Introduction date, latest release and number of entries have been determined from information on the web page of the database, if it is still extant, and publications about the database. Databases marked with † are defunct. I have defined the number of entries as the number of groupings, alignments or models in the database.

of manual databases suggests that human knowledge is an essential component of the value provided by protein family databases, and synthesising that component has proved elusive.

In a curated database, human curators are required to construct models and to annotate families with relevant information such as literature references, functional or structural information. Replacing a skilled and knowledgable person with a rack of computers, freeing the person to work on other problems not yet possible to automate, is an enticing idea. Using clustering algorithms has enabled automated construction of models, but I am unaware of any database capable of automatically annotating novel groupings of proteins which it has found. It seems that such a complex association of sequence to structure and function is still the preserve of humans. And, as previously noted, groupings found by automated databases can be spurious,

outweighing the benefit of genuine novel groupings that may be found (Finn, Coggill *et al.*, 2015).

If people are required to construct a high-quality protein family database, it is clear that a limiting factor on the addition of new families to the database will be the annotator-hours expended on it. Human time as a limiting factor is a familiar problem, and an approach to the mitigation of this factor investigated in a number of applications is *crowdsourcing*. Crowdsourcing refers to breaking up a workload that might be assigned to one person, or a small group, into many smaller tasks, and distributing these tasks to a large number of people (Good and Su, 2013). An additional characteristic of the strategy is that the individuals completing the tasks are compensated on the basis of its completion rather than employed on an ongoing basis, or are not compensated at all, and complete it purely out of altruism. The Internet is the usual distribution mechanism for the tasks. If crowdsourcing can be applied to protein family creation and annotation, we hypothesise that this will allow protein family databases to grow more rapidly without entailing additional costs.

Pfam

Pfam is a database of protein families. The Pfam 1.0 release in 1997 provided models for 175 families (Sonnhammer *et al.*, 1997). The current release contains 16,712 entries (Finn, Coggill *et al.*, 2015). Each of these entries is described by a *seed alignment*, which is used to generate a profile HMM, using the HMMER software package (Eddy, 1998). This model is then used to query the protein sequence database UniProtKB, and significant matches for the model are recorded. In Pfam, no region should be matched by more than one model (Bateman, Birney *et al.*, 1999). Prior to each release of the database, overlap analysis is performed to determine if any residue in UniProtKB is matched by more than one model. This overlap criterion is an important quality control mechanism (Sonnhammer *et al.*, 1997). The most recent releases of Pfam have relaxed the overlap criteria slightly, to allow short areas of overlap which do not affect a high proportion of the family members (Finn, Coggill *et al.*, 2015).

When overlaps are found, a curator will either adjust the significance threshold for one of the overlapping models, or adjust the start or end coordinates of the seed alignment. Alternatively, they may conclude that the two families are evolutionarily related, and should be grouped together. In some cases, the families can be merged together, by creating a profile HMM from the seed sequences of both. In other cases, either where the curator wishes to maintain separate entries for the families, or if it isn't possible to create a profile HMM which matches all members of both families, a clan can be created. In Pfam, families which are both in the same clan are allowed to overlap. For classification purposes, when a protein region matches two families in

the same clan, the region is assigned to the family with the higher bit score for the region (Finn, Coggill *et al.*, 2015).

Since 2011, Pfam has used Wikipedia[3] to crowdsource the curation of entries (Finn, Bateman *et al.*, 2014). In cases where there is an appropriate Wikipedia article, the corresponding Pfam entry is linked to it, and the content of the article is displayed to users of Pfam as the main description of the family. The 'basic' textual annotation written by Pfam curators is retained, but is less emphasised. In other cases, basic 'stub' Wikipedia articles are created for entries which don't have an existing Wikipedia article. This allows the community to improve the annotation of the entry by editing the Wikipedia article. This system was pioneered in the Rfam RNA family database (Gardner *et al.*, 2011). In addition to the improvements to annotation, linking the database entry to the Wikipedia entry increases web traffic to the database.

1.2.2. Hidden Markov models

Hidden Markov models are a statistical tool for modelling a dynamic system, and are a type of Markov model (Rabiner, 1990). The simplest type of Markov model is a Markov chain. In a Markov chain, the system is described by a number of states, with a transition probability between each state. The Markov assumption is that the probability of transitioning to a particular state depends only on what the n previous states were, for a Markov chain of order n. It is always possible to decompose a Markov chain of any order to a first order Markov chain. An example of a dynamic system is the weather. By making the assumption that future weather depends entirely on the current weather, we can model it using a first order Markov chain.

Given the weather model in figure 1.3, we can answer the questions like 'what is the likelihood of cloud in the next time step', 'what is the likelihood that it keeps raining', or 'how likely is it that there will be some sun in the next four time steps'.

In an HMM, this idea is extended to dynamic systems where the current state of the system cannot be observed. In addition to the transition probabilities between the (now hidden) states of the system, each hidden state also has a set of emission probabilities, which encode the likelihood of a particular output of the system being observed. Using the weather example, if we work in a windowless office, the current state of the weather is no longer possible to observe. However, we could observe various pieces of evidence about what the current weather is, such as our colleague's attire. Given such observations, we could model weather as in figure 1.4. An HMM provides a toolkit of algorithms for reasoning about the system. The Baum-Welch algorithm allows us to estimate the transition and emission probabilities. The Viterbi algorithm

[3] `http://en.wikipedia.org/`

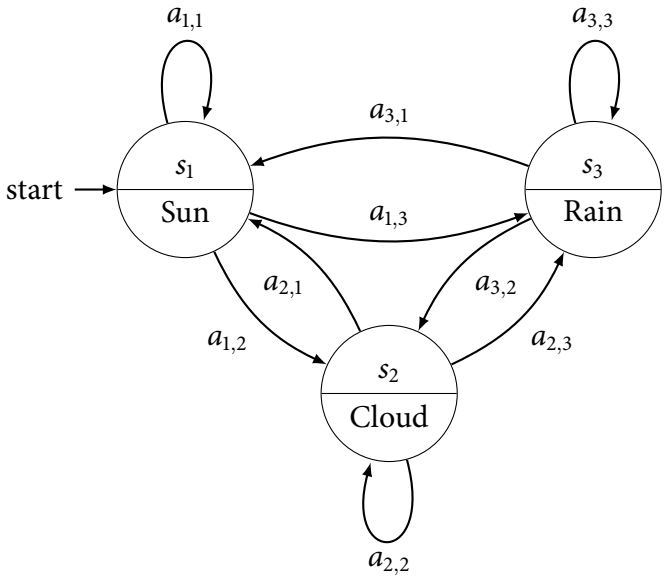


Figure 1.3. A simple Markov chain for the weather. At each time step, the weather has probability $a_{x,y}$ of moving from its current state x to successor state y.

can be used to determine the most likely progression of states (Rabiner, 1990). In this case, the most likely past and current state of the weather. The forward algorithm can tell us how likely a particular sequence of observations are, given any possible weather.

An early problem to which HMMs were applied was speech recognition. Speech recognition is an example of a decoding problem. The observable information is a noisy waveform, and the task is to attempt to 'decode' the waveform into text. An HMM can be quite naturally applied in this case. The waveform is modelled as an *emission*, which is *generated* by the text. For this reason, the application of the Viterbi algorithm to identify the most likely progression of hidden states is often called *decoding*.

In biology, sequence analysis provides many tasks which can be formulated as decoding problems, enabling the use of the HMM toolkit of algorithms. The recognition of protein families is one such task. In a protein sequence, there may be one or more regions which are similar to other members of a protein family, due to their common descent from an ancestral protein sequence. We are unable to know for certain where the boundaries of these homologous regions are, but we can infer it from sequence conservation. If two regions are very similar in sequence, then this is evidence that they are homologous. A fundamental technique for identifying homology between two proteins is to apply an alignment algorithm, such as Smith-Waterman. The regions of the two protein sequences which are aligned to each other, the match regions, may be homologues.

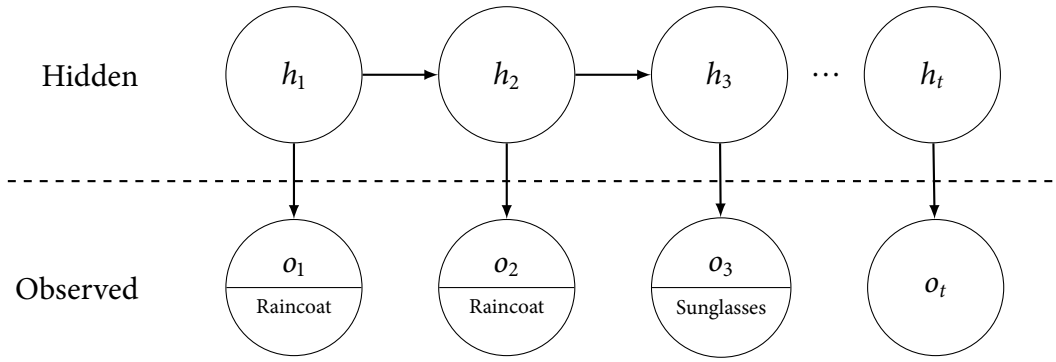


Figure 1.4. The progression of state through time for an HMM of the weather. Rather than observing the state, h, of the weather, we can only observe the consequences of the weather, that is, the attire of our colleague, o. Using the HMM toolkit, we can reason about the possible hidden state of the system.

Using this technique, we could build a list of proteins which we believe have homologous regions, that is, the members of a protein family. But what if we want to determine whether a particular sequence is likely to be a member of the family? Perhaps we could try to align the sequence to every single member of the family. But, looking at the entire family we can also observe that some regions of a protein are more likely to be conserved than others. In other words, the likelihood of substitution depends not only on the identity of the two residues, but also on their position along the protein sequence. They are *position dependent*. Furthermore, some regions appear more amenable to insertions and deletions. Classical alignment algorithms such as Smith-Waterman are *position independent*, so they are not able to use this information when performing alignment.

An HMM allows us to encode all of this information. We can model the residues as the emission of the system, which is generated by the protein family (Durbin *et al.*, 1998). Such a model is called a profile HMM. The profile HMM was introduced by Krogh *et al.* (1994). In figure 1.5, the hidden states for a profile HMM based upon the model of Krogh *et al.* (1994) is shown. This model is largely conserved in more recent profile HMM approaches. Each match state m will have a high likelihood of emitting residues which have been observed at this position within family members, but with a non-zero probability of emitting any residue, to account for substitution mutations. The transition probabilities to insert and delete states can be set either using position independent probabilities derived from the observed rate of mutation between the different amino acids, encoded in a point accepted mutation (PAM) matrix, or using position dependent probabilities derived from the observed insertions amongst family members, as can the emission probabilities for the insert states i. It's interesting to note that the first PAM matrix

was distributed by Margaret Oakley Dayhoff in the *Atlas*. Having created the first database of proteins, she was able to analyse computationally the likelihood of mutations between amino acids by comparing the sequences of closely related proteins (Dayhoff and Schwartz, 1978; Hagen, 2011).

To determine the most likely alignment of a protein sequence with a protein family, we set the observed output of the family's profile HMM to the protein sequence, and then decode the most likely hidden state of the model with the Viterbi algorithm. The sequence of hidden insert, match and delete states gives the most likely alignment. As noted by Jurafsky and Martin (2009), the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) was a later, independent discovery of the Viterbi algorithm. As the Smith-Waterman algorithm (Smith and Waterman, 1981) is a variation of the Needleman-Wunsch, it too is equivalent to the Viterbi algorithm (Krogh *et al.*, 1994). The key difference is that decoding a profile HMM allows the calculation of a probability for the observed sequence, whereas the conventional alignment algorithms will produce a 'score'.

As noted above, the match states for a protein family profile HMM should emit different amino acids in proportion with the likelihood of a protein within the family having that amino acid at the corresponding position. For a given family, it is necessary to estimate the observation probabilities for every amino acid for each match state. A very naive approach would be to take the alignment of all known members of the family, and assume that the distribution of possible amino acids at each position represented the population distribution for the entire family (Durbin *et al.*, 1998). Of course, this would lead to zero probability being assigned to the observation of any amino acid which wasn't seen at a given position in the sequence of any of the known family members. Thus, the emission probability of a sequence containing such an amino acid would be zero. A model using probabilities estimated in this way would be unable to identify family members which had even very minor mutations which hadn't been observed in any known members.

Methods for reassigning probability mass from observed values to unobserved values are called 'smoothing' (Jurafsky and Martin, 2009). The simplest method, and for this application, the most frequently discussed, is Laplace smoothing (Durbin *et al.*, 1998). In Laplace smoothing, a constant, called a 'pseudocount', is added to the number of observations of each possible value (Durbin *et al.*, 1998). So for any amino acid which wasn't observed in a position, the smoothed count would be the pseudocount. While this method removes the undesirable possibility of any alignment being scored zero, it doesn't account for the fact that some amino acid substitutions are more common than others. On the other hand, if more evidence is available, the pseudocounts have less of an effect on the probability distribution, reflecting the fact that

as more proteins are added to the alignment, we can become more confident that it represents the true distribution of the family (Sjölander *et al.*, 1996).

A second method for estimating the emission probabilities is to incorporate information about mutation rates from substitution matrices. Such matrices are derived from the alignments of large numbers of sequences, and provide a non-zero estimate for the probability of possible substitution. However, due to the structure of a particular protein, the probability of substitution for a particular pair may be different from that of another protein (Sjölander *et al.*, 1996).

In fact, it would be better to integrate information from the alignment, but also from prior knowledge of amino acid substitution. A Bayesian approach to the problem would give more weight to prior information when we have an uncertain estimate (Durbin *et al.*, 1998). That is, if we have an alignment with few proteins, we should trust out prior knowledge more. As members are added, the probability estimates should converge on the distribution implied by known members (J. G. Henikoff and S. Henikoff, 1996). A method for doing this was proposed by Brown *et al.* (1993). They assume that the amino acid counts for each position in the protein family alignment has been sampled from an unknown mixture of Dirichlet distributions. The priors for this distribution are estimated from a database of alignments of homologous sequences. The use of a mixture distribution accounts for the fact that in different contexts, different amino acids are more likely (Durbin *et al.*, 1998; Sjölander *et al.*, 1996). By using such methods, profile HMMs are able to identify distant homologues, using relatively few examples of a protein family, a particularly important consideration at the time of their development, when few example sequences were available.

To find the likelihood that the protein sequence is homologous with the protein family, we must sum of the probability of the protein sequence being emitted over all possible sequences of hidden states. That is, the probability of every possible alignment given the protein sequence. This is one of the fundamental advantages of profile HMM based methods over the traditional alignment algorithms: It is mathematically sound to sum over probabilities, whereas the same is not true of an arbitrary score (Eddy, 2009). Therefore, the traditional alignment algorithms make the assumption that the score of the optimal alignment for a pair of sequences is directly proportional to the likelihood that the pair are homologous. In contrast HMM based methods can sum over the probabilities of all possible alignments of the pair. This summing is accomplished with the forward algorithm. But it is well known that Smith-Waterman is too slow to feasibly identify homologues of a query sequence in a large sequence database. If the Viterbi algorithm is equivalent to Smith-Waterman, how can we hope to use profile HMM based methods when dealing with millions of sequences?

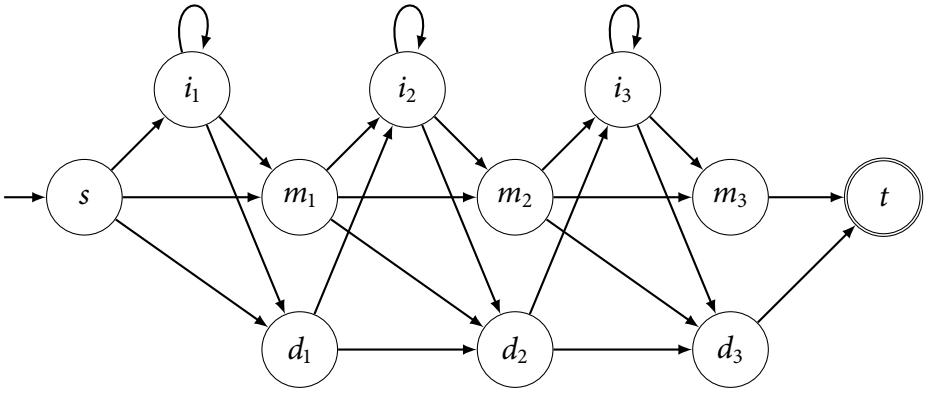


Figure 1.5. A Markov chain for a protein sequence of length three, with match states m, insert states i, delete states d and start and end states s and t. This model forms the hidden state of a profile HMM.

HMMER

HMMER is a software package developed by Eddy (1998) for performing sequence homology search. Eddy's work over the past twenty years has focussed on the exploitation of HMMs to make homology search more sensitive and accurate. Since the beginning of the 21st century, the decreased cost of compute resources, and algorithmic developments have made it feasible to use HMM based methods with large numbers of sequences. Previously, only less accurate but faster heuristic methods like BLAST were fast enough to feasibly search a sequence database of millions using a profile HMM. Eddy notes that BLAST (Altschul, Gish *et al.*, 1990) is intended to approximate the scores produced by Smith-Waterman, but uses heuristics to avoid producing a full alignment for every single sequence pair (Eddy, 1998). He developed HMMER as an equivalent to BLAST for HMM based methods.

The HMMER package emerged in 1995, at around the same time that Richard Hughey, Kevin Karplus and Anders Krogh released their own software, Sequence Alignment and Modelling Software System (SAM), for performing similar functions (Hughey and Krogh, 1996). With their colleagues in David Haussler's group, they carried out much of the original research into the use of profile HMMs for homology search, discussed in the previous section, including the model architecture, and the use of Dirichlet mixture distribution priors (Brown *et al.*, 1993; Sjölander *et al.*, 1996). Eddy integrated many of their algorithms and ideas into his own software, and HMMER eventually outlived SAM (Eddy and Wheeler, 2015).

By filtering sequences using a simplified profile HMM, recent versions of HMMER are able to avoid computing the costly full Viterbi alignment and forward scores for the vast majority of sequences in the database. Thus, HMMER has performance comparable with BLAST, but with

the benefits of a statistically grounded HMM. Most notably, through the use of the forward algorithm, HMMER is able to calculate the probability of all possible alignments of a protein sequence to a sequence profile, instead of using the probability of the most likely alignment as an estimate thereof. By finding this missing probability mass, HMMER is able to find more distant homologues, which would otherwise be under the significance threshold (Eddy, 2009).

HMMER provides a number of tools based around the use of profile HMMs to query sequence databases. To perform a sequence similarity search, the *phmmer* program is used. It converts the query sequence to a profile HMM by converting the residue substitution scores from the BLOSUM62 matrix to probabilities, in a position independent way (Eddy and Wheeler, 2015). Indels are accounted for using a position independent gap-open and gap-extend probability. This model is then used to find the probability that a particular sequence in the database is a homologue of the query sequence (Eddy, 1998; Finn, Clements *et al.*, 2011).

The *hmmbuild* program constructs a profile HMM from a multiple sequence alignment. This tool is used by Pfam curators to produce a family profile HMM from an alignment of seed sequences. The *hmmsearch* program uses a profile HMM to query a protein sequence database. The listed members of a Pfam family are the protein sequences which exceed the family's significance threshold for an *hmmsearch* query against the UniProt Reference Proteomes sequence database (Chen *et al.*, 2011).

The *jackhmmer* program performs the same function as *phmmer*, but iteratively, and is analogous to PSI-BLAST (Altschul, Madden *et al.*, 1997; Eddy and Wheeler, 2015). Like PSI-BLAST, an initial set of sequence similarity search results are aligned, and then used to identify further homologues. The first published profile HMM based iterative searching strategy was SAM-T98, and this was followed by incremental improvements SAM-T99 and target2k (Karplus *et al.*, 1998). Unlike previous methods, *jackhmmer* uses profile HMM searches to query the entire sequence database, unlike target2k, which uses BLAST to construct a 'subdatabase' (Johnson *et al.*, 2010). In the initial iteration, *phmmer* is used to construct a result set of homologues with the query sequence. These results are used to construct an alignment and from this a profile HMM. This model is then used to query the database, as *hmmsearch* does, and construct a new result set. The process is repeated for a fixed number of iterations, or until convergence (an iteration in which no region either leaves or joins the result set). This process is able to detect more remote homologues than a single iteration of *phmmer* (Johnson *et al.*, 2010). Both of these tools are available as web services, provided by the EMBL-EBI.[4]

In this project, **I will develop a framework to enable a community annotated protein family database** which retains the desirable quality control of a curated database, while enabling a

[4] `http://www.ebi.ac.uk/Tools/hmmer/`

larger area of protein space to be covered. This kind of database would use information supplied by the community, algorithmically quality controlled, and with curators maintaining ultimate control over inclusion. Such a database could either be constructed *de novo* or based upon an existing curated database. We have identified four enabling aspects for such a system:

Coverage and Quality Algorithmically identifying which families to add to the database. That is, those which increase coverage without sacrificing quality.

User Contributions Determine how users can contribute their expertise with minimal training in the domain of protein classification.

Enrichment with Relevant Literature Use modern natural language processing techniques to extract relevant information from the literature to compensate for reduced curator input.

Curation users Evaluate the use of the system by professional curators, to see if this improves upon existing workflows.

1.3. References

Abbott, A. 'Swiss databank to start charging for use'. In: *Nature* 394.6690 (July 1998), p. 214. ISSN: 0028-0836. DOI: 10.1038/28249. PUBMED: 9685150.

Altschul, S. F., T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman. 'Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.' In: *Nucleic acids research* 25.17 (Sept. 1997), pp. 3389–3402. ISSN: 0305-1048. PUBMED: 9254694.

Altschul, S. F., W. Gish, W. Miller, E. W. Myers and D. J. Lipman. 'Basic local alignment search tool'. In: *Journal of Molecular Biology* 215.3 (Oct. 1990), pp. 403–410. ISSN: 00222836. DOI: 10.1016/S0022-2836(05)80360-2. PUBMED: 2231712.

Apweiler, R., V. Junker, A. Gateau, C. O'Donovan, F. Lang and A. Bairoch. 'New developments in linking of biological databases and computer-generation of annotation: SWISS-PROT and its computer-annotated supplement TREMBL'. In: *Bioinformatics. GCB 1996*. Ed. by R. Hofestädt, T. Lengauer, M. Löffler and D. Schomburg. Lecture Notes in Computer Science, vol 1278. Springer, Berlin, Heidelberg, 1996, pp. 44–51. ISBN: 978-3-540-69524-0. DOI: 10.1007/BFb0033202.

Attwood, T. K., M. D. Croning, D. R. Flower, A. P. Lewis, J. E. Mabey, P. Scordis, J. N. Selley and W. Wright. 'PRINTS-S: the database formerly known as PRINTS.' In: *Nucleic acids research* 28.1 (Jan. 2000), pp. 225–227. ISSN: 0305-1048. DOI: 10.1093/nar/28.1.225. PUBMED: 10592232.

Attwood, T. K., A. Coletta, G. Muirhead, A. Pavlopoulou, P. B. Philippou, I. Popov, C. Romá-Mateo, A. Theodosiou and A. L. Mitchell. 'The PRINTS database: a fine-grained protein sequence annotation and analysis resource–its status in 2012.' In: *Database* 2012 (Jan. 2012). ISSN: 1758-0463. DOI: 10.1093/database/bas019. PUBMED: 22508994.

Bairoch, A. and B. Boeckmann. 'The SWISS-PROT protein sequence data bank'. In: *Nucleic Acids Research* 19.suppl (Apr. 1991), pp. 2247–2249. ISSN: 1362-4962. DOI: 10.1093/nar/19.suppl.2247. PUBMED: 2041811.

Barker, W., J. S. Garavelli, D. H. Haft, L. T. Hunt, C. R. Marzec, B. C. Orcutt, G. Y. Srinivasarao, L.-S. L. Yeh, R. S. Ledley, H.-W. Mewes, F. Pfeiffer and A. Tsugita. 'The PIR-International Protein Sequence Database'. In: *Nucleic acids research* 26.1 (Jan. 1998), pp. 27–32. ISSN: 1362-4962. DOI: 10.1093/nar/26.1.27. PUBMED: 9847137.

Bateman, A., E. Birney, R. Durbin, S. R. Eddy, R. D. Finn and E. L. L. Sonnhammer. 'Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins'. In: *Nucleic acids research* 27.1 (Jan. 1999), pp. 260–262. ISSN: 0305-1048. DOI: 10.1093/nar/27.1.260. PUBMED: 9847196.

Bateman, A. and E. Birney. 'Searching databases to find protein domain organization'. In: *Analysis of Amino Acid Sequences*. Ed. by P. Bork. Vol. 54. Advances in Protein Chemistry. Elsevier, 2000, pp. 137–157. ISBN: 9780120342549. DOI: 10.1016/S0065-3233(00)54005-4.

Bateman, A., L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Studholme, C. Yeats and S. R. Eddy. 'The Pfam protein families database.' In: *Nucleic acids research* 32.Database issue (Jan. 2004), pp. D138–141. ISSN: 1362-4962. DOI: 10.1093/nar/gkh121. PUBMED: 14681378.

Brown, M., R. Hughey, A. Krogh, I. S. Mian, K. Sjölander and D. Haussler. 'Using Dirichlet mixture priors to derive hidden Markov models for protein families.' In: *Proceedings. International Conference on Intelligent Systems for Molecular Biology* 1 (1993), pp. 47–55. ISSN: 1553-0833. PUBMED: 7584370.

Bru, C. 'The ProDom database of protein domain families: more emphasis on 3D'. In: *Nucleic acids research* 33.Database issue (Jan. 2005), pp. D212–D215. ISSN: 1362-4962. DOI: 10.1093/nar/gki034. PUBMED: 15608179.

Butler, D. 'Bidding heats up for protein database'. In: *Nature* 381.6581 (May 1996), p. 266. ISSN: 0028-0836. DOI: 10.1038/381266b0. PUBMED: 8692257.

– 'NIH pledges cash for global protein database'. In: *Nature* 419.6903 (Sept. 2002), p. 101. ISSN: 0028-0836. DOI: 10.1038/419101a. PUBMED: 12226627.

Chen, C., D. A. Natale, R. D. Finn, H. Huang, J. Zhang, C. H. Wu and R. Mazumder. 'Representative proteomes: a stable, scalable and unbiased proteome set for sequence analysis and

functional annotation.' In: *PloS one* 6.4 (Apr. 2011). Ed. by J. D. Hoheisel, e18910. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0018910. PUBMED: 21556138.

Crick, F. H. C. 'On protein synthesis.' In: *Symposia of the Society for Experimental Biology* 12 (Jan. 1958), pp. 138–163. ISSN: 0081-1386. PUBMED: 13580867.

Dayhoff, M. O. 'The origin and evolution of protein superfamilies.' In: *Federation proceedings* 35.10 (Aug. 1976), pp. 2132–2138. ISSN: 0014-9446. PUBMED: 181273.

Dayhoff, M. O. 'Computer analysis of protein sequences'. In: *Computers in Life Science Research*. Boston, MA: Springer US, 1974, pp. 9–14. DOI: 10.1007/978-1-4757-0546-1_3.

Dayhoff, M. O., W. C. Barker and P. J. McLaughlin. 'Inferences from protein and nucleic acid sequences: Early molecular evolution, divergence of kingdoms and rates of change'. In: *Origins of Life* 5.3-4 (July 1974), pp. 311–330. ISSN: 0302-1688. DOI: 10.1007/BF01207633. PUBMED: 4370101.

Dayhoff, M. O. and R. M. Schwartz. 'A model of evolutionary change in proteins'. In: *Atlas of Protein Sequence and Structure (Vol 5, Supplement 3)*. National Biomedical Research Foundation, 1978. Chap. 22, pp. 345–352. ISBN: 0-912466-07-3.

Dayhoff, M. O. *Atlas of Protein Sequence and Structure*. Vol. 5. Silver Spring, MD: National Biomedical Research Foundation, 1972. ISSN: 0572-435X.

Durbin, R., S. Eddy, A. Krogh and G. Mitchison. *Biological sequence analysis*. 1998. ISBN: 0-521-62971-3. DOI: 10.1017/CBO9780511790492.

Eddy, S. R. 'Profile hidden Markov models'. In: *Bioinformatics* 14.9 (Oct. 1998), pp. 755–763. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/14.9.755. PUBMED: 9918945.

Eddy, S. R. 'A new generation of homology search tools based on probabilistic inference.' In: *Genome informatics*. Vol. 23. Oct. 2009, pp. 205–211. DOI: 10.1142/9781848165632_0019. PUBMED: 20180275.

Eddy, S. R. and T. J. Wheeler. *HMMER User's Guide*. Feb. 2015. URL: eddylab.org/software/hmmer3/3.1b2/Userguide.pdf.

Finn, R. D., A. Bateman, J. Clements, P. Coggill, R. Y. Eberhardt, S. R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. L. L. Sonnhammer, J. Tate and M. Punta. 'Pfam: the protein families database.' In: *Nucleic acids research* 42.D1 (Jan. 2014), pp. D222–230. ISSN: 1362-4962. DOI: 10.1093/nar/gkt1223. PUBMED: 24288371.

Finn, R. D., J. Clements and S. R. Eddy. 'HMMER web server: interactive sequence similarity searching.' In: *Nucleic acids research* 39.Web Server issue (July 2011), W29–W37. ISSN: 1362-4962. DOI: 10.1093/nar/gkr367. PUBMED: 21593126.

Finn, R. D., P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, G. A. Salazar, J. Tate and A. Bateman. 'The Pfam protein

families database: towards a more sustainable future.' In: *Nucleic acids research* 44.D1 (Jan. 2015), pp. D279–D285. ISSN: 1362-4962. DOI: 10.1093/nar/gkv1344. PUBMED: 26673716.

Finn, R. D., J. Mistry, B. Schuster-Böckler, S. Griffiths-Jones, V. Hollich, T. Lassmann, S. Moxon, M. Marshall, A. Khanna, R. Durbin, S. R. Eddy, E. L. L. Sonnhammer and A. Bateman. 'Pfam: clans, web tools and services.' In: *Nucleic acids research* 34.Database issue (Jan. 2006), pp. D247–D251. ISSN: 1362-4962. DOI: 10.1093/nar/gkj149. PUBMED: 16381856.

Gardner, P. P., J. Daub, J. Tate, B. L. Moore, I. H. Osuch, S. Griffiths-Jones, R. D. Finn, E. P. Nawrocki, D. L. Kolbe, S. R. Eddy and A. Bateman. 'Rfam: Wikipedia, clans and the "decimal" release.' In: *Nucleic acids research* 39.Database issue (Jan. 2011), pp. D141–D145. ISSN: 1362-4962. DOI: 10.1093/nar/gkq1129. PUBMED: 21062808.

George, D. G., W. C. Barker and L. T. Hunt. 'The protein identification resource (PIR).' In: *Nucleic acids research* 14.1 (Jan. 1986), pp. 11–5. ISSN: 0305-1048. PUBMED: 3945547.

Good, B. M. and A. I. Su. 'Crowdsourcing for bioinformatics.' In: *Bioinformatics* 29.16 (Aug. 2013), pp. 1925–1933. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt333. PUBMED: 23782614. arχiv: 1302.6667.

Gracy, J. and P. Argos. 'Automated protein sequence database classification. I. Integration of compositional similarity search, local similarity search, and multiple sequence alignment.' In: *Bioinformatics* 14.2 (Jan. 1998), pp. 164–173. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/14.2.164. PUBMED: 9545449.

– 'Automated protein sequence database classification. II. Delineation of domain boundaries from sequence similarities.' In: *Bioinformatics* 14.2 (Jan. 1998), pp. 174–187. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/14.2.174. PUBMED: 9545450.

Haft, D. H., J. D. Selengut, R. A. Richter, D. Harkins, M. K. Basu and E. Beck. 'TIGRFAMs and Genome Properties in 2013.' In: *Nucleic acids research* 41.Database issue (Jan. 2013), pp. D387–95. ISSN: 1362-4962. DOI: 10.1093/nar/gks1234. PUBMED: 23197656.

Hagen, J. B. 'The origin and early reception of sequence databases.' In: *Data Mining in Proteomics*. Ed. by M. Hamacher, M. Eisenacher and C. Stephan. Vol. 696. Methods in Molecular Biology. Humana Press, Jan. 2011, pp. 61–77. DOI: 10.1007/978-1-60761-987-1_4. PUBMED: 21063941.

Henikoff, J. G., E. A. Greene, S. Pietrokovski and S. Henikoff. 'Increased coverage of protein families with the blocks database servers.' In: *Nucleic acids research* 28.1 (Jan. 2000), pp. 228–230. ISSN: 0305-1048. DOI: 10.1093/nar/28.1.228. PUBMED: 10592233.

Henikoff, J. G. and S. Henikoff. 'Using substitution probabilities to improve position-specific scoring matrices'. In: *Bioinformatics* 12.2 (Apr. 1996), pp. 135–143. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/12.2.135.

Hughey, R. and A. Krogh. 'Hidden Markov models for sequence analysis: extension and analysis of the basic method'. In: *Bioinformatics* 12.2 (Apr. 1996), pp. 95–107. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/12.2.95.

Johnson, L. S., S. R. Eddy and E. Portugaly. 'Hidden Markov model speed heuristic and iterative HMM search procedure.' In: *BMC bioinformatics* 11.431 (Jan. 2010). ISSN: 1471-2105. DOI: 10.1186/1471-2105-11-431. PUBMED: 20718988.

Jurafsky, D. and J. H. Martin. *An Introduction to Natural Language Processing , Computational Linguistics, and Speech Recognition*. Pearson, 2009. ISBN: 9780131873216.

Karplus, K., C. Barrett and R. Hughey. 'Hidden Markov models for detecting remote protein homologies'. In: *Bioinformatics* 14.10 (Nov. 1998), pp. 846–856. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/14.10.846.

Krogh, A., I. S. Mian and D. Haussler. 'A hidden Markov model that finds genes in E. coli DNA.' In: *Nucleic acids research* 22.22 (Nov. 1994), pp. 4768–4778. ISSN: 0305-1048. DOI: 10.1093/nar/22.22.4768. PUBMED: 7984429.

Letunic, I. and P. Bork. '20 years of the SMART protein domain annotation resource'. In: *Nucleic acids research* 46.D1 (Jan. 2018), pp. D493–D496. ISSN: 0305-1048. DOI: 10.1093/nar/gkx922. PUBMED: 29040681.

Meinel, T., A. Krause, H. Luz, M. Vingron and E. Staub. 'The SYSTERS Protein Family Database in 2005.' In: *Nucleic acids research* 33.Database issue (Jan. 2005), pp. D226–229. ISSN: 1362-4962. DOI: 10.1093/nar/gki030. PUBMED: 15608183.

Mi, H., S. Poudel, A. Muruganujan, J. T. Casagrande and P. D. Thomas. 'PANTHER version 10: expanded protein families and functions, and analysis tools.' In: *Nucleic acids research* 44.D1 (Nov. 2015), pp. D336–342. ISSN: 1362-4962. DOI: 10.1093/nar/gkv1194. PUBMED: 26578592.

Mitchell, A., H.-Y. Chang, L. Daugherty, M. Fraser, S. Hunter, R. Lopez, C. McAnulla, C. McMenamin, G. Nuka, S. Pesseat, A. Sangrador-Vegas, M. Scheremetjew, C. Rato, S.-Y. Yong, A. Bateman, M. Punta, T. K. Attwood, C. J. A. Sigrist, N. Redaschi, C. Rivoire, I. Xenarios, D. Kahn, D. Guyot, P. Bork, I. Letunic, J. Gough, M. Oates, D. Haft, H. Huang, D. A. Natale, C. H. Wu, C. Orengo, I. Sillitoe, H. Mi, P. D. Thomas and R. D. Finn. 'The InterPro protein families database: the classification resource after 15 years.' In: *Nucleic acids research* 43.Database issue (Jan. 2015), pp. D213–D221. ISSN: 1362-4962. DOI: 10.1093/nar/gku1243. PUBMED: 25428371.

Needleman, S. B. and C. D. Wunsch. 'A general method applicable to the search for similarities in the amino acid sequence of two proteins'. In: *Journal of Molecular Biology* 48.3 (Mar. 1970), pp. 443–453. ISSN: 0022-2836. DOI: 10.1016/0022-2836(70)90057-4.

Oates, M. E., J. Stahlhacke, D. V. Vavoulis, B. Smithers, O. J. L. Rackham, A. J. Sardar, J. Zaucha, N. Thurlby, H. Fang and J. Gough. 'The SUPERFAMILY 1.75 database in 2014: a doubling of data.' In: *Nucleic acids research* 43.D1 (Jan. 2015), pp. D227–D233. ISSN: 1362-4962. DOI: 10.1093/nar/gku1041. PUBMED: 25414345.

Pedruzzi, I., C. Rivoire, A. H. Auchincloss, E. Coudert, G. Keller, E. de Castro, D. Baratin, B. A. Cuche, L. Bougueleret, S. Poux, N. Redaschi, I. Xenarios and A. Bridge. 'HAMAP in 2015: updates to the protein family classification and annotation system.' In: *Nucleic acids research* 43.D1 (Jan. 2015), pp. D1064–D1070. ISSN: 1362-4962. DOI: 10.1093/nar/gku1002. PUBMED: 25348399.

Portugaly, E., N. Linial and M. Linial. 'EVEREST: a collection of evolutionary conserved protein domains.' In: *Nucleic acids research* 35.Database issue (Jan. 2007), pp. D241–D246. ISSN: 1362-4962. DOI: 10.1093/nar/gkl850. PUBMED: 17099230.

Rabiner, L. R. *A tutorial on hidden Markov models and selected applications in speech recognition*. Ed. by A. Waibel and K.-F. Lee. Morgan Kaufmann Publishers, 1990, pp. 267–296. ISBN: 1558601244. URL: https://dl.acm.org/citation.cfm?id=108253.

Sanger, F. and E. O. P. Thompson. 'The amino-acid sequence in the glycyl chain of insulin. I. The identification of lower peptides from partial hydrolysates.' In: *The Biochemical journal* 53.3 (Feb. 1953), pp. 353–366. ISSN: 0264-6021. PUBMED: 13032078.

– 'The amino-acid sequence in the glycyl chain of insulin. II. The investigation of peptides from enzymic hydrolysates.' In: *The Biochemical journal* 53.3 (Feb. 1953), pp. 366–374. ISSN: 0264-6021. PUBMED: 13032079.

Sanger, F. and H. Tuppy. 'The amino-acid sequence in the phenylalanyl chain of insulin. 1. The identification of lower peptides from partial hydrolysates.' In: *The Biochemical journal* 49.4 (Sept. 1951), pp. 463–481. ISSN: 0264-6021. PUBMED: 14886310.

– 'The amino-acid sequence in the phenylalanyl chain of insulin. 2. The investigation of peptides from enzymic hydrolysates.' In: *The Biochemical journal* 49.4 (Sept. 1951), pp. 481–490. ISSN: 0264-6021. PUBMED: 14886311.

Servant, F., C. Bru, S. Carrère, E. Courcelle, J. Gouzy, D. Peyruc and D. Kahn. 'ProDom: automated clustering of homologous domains.' In: *Briefings in bioinformatics* 3.3 (Jan. 2002), pp. 246–251. ISSN: 1467-5463. DOI: 10.1093/bib/3.3.246. PUBMED: 12230033.

Sidman, K. E., D. G. George, W. C. Barker and L. T. Hunt. 'The protein identification resource (PIR).' In: *Nucleic acids research* 16.5 (Mar. 1988), pp. 1869–71. ISSN: 0305-1048. PUBMED: 3353227.

Sigrist, C. J. A., E. de Castro, L. Cerutti, B. A. Cuche, N. Hulo, A. Bridge, L. Bougueleret and I. Xenarios. 'New and continuing developments at PROSITE.' In: *Nucleic acids research* 41.Data-

base issue (Jan. 2013), pp. D344–D347. ISSN: 1362-4962. DOI: 10.1093/nar/gks1067. PUBMED: 23161676.

Sjölander, K., K. Karplus, M. Brown, R. Hughey, A. Krogh, I. S. Mian and D. Haussler. 'Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology.' In: *Computer applications in the biosciences : CABIOS* 12.4 (Aug. 1996), pp. 327–45. ISSN: 0266-7061. PUBMED: 8902360.

Smith, T. and M. Waterman. 'Identification of common molecular subsequences'. In: *Journal of Molecular Biology* 147.1 (Mar. 1981), pp. 195–197. ISSN: 0022-2836. DOI: 10.1016/0022-2836(81)90087-5.

Sonnhammer, E. L., S. R. Eddy and R. Durbin. 'Pfam: A comprehensive database of protein domain families based on seed alignments'. In: *Proteins* 28.3 (July 1997), pp. 405–420. ISSN: 08873585. DOI: 10.1002/(SICI)1097-0134(199707)28:3<405::AID-PROT10>3.0.CO;2-L. PUBMED: 9223186.

Srinivasarao, G. Y., L. S. Yeh, C. R. Marzec, B. C. Orcutt, W. C. Barker and F. Pfeiffer. 'Database of protein sequence alignments: PIR-ALN.' In: *Nucleic acids research* 27.1 (Jan. 1999), pp. 284–285. ISSN: 0305-1048. DOI: 10.1093/nar/27.1.284. PUBMED: 9847202.

Strasser, B. J. 'Collecting, Comparing, and Computing Sequences: The Making of Margaret O. Dayhoff's Atlas of Protein Sequence and Structure, 1954–1965'. In: *Journal of the history of biology* 43.4 (Dec. 2010), pp. 623–660. ISSN: 0022-5010. DOI: 10.1007/s10739-009-9221-0. PUBMED: 20665074.

The UniProt Consortium. 'UniProt: the universal protein knowledgebase'. In: *Nucleic acids research* 45.D1 (Jan. 2017), pp. D158–D169. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1099. PUBMED: 27899622.

van der Lee, R., M. Buljan, B. Lang, R. J. Weatheritt, G. W. Daughdrill, A. K. Dunker, M. Fuxreiter, J. Gough, J. Gsponer, D. T. Jones, P. M. Kim, R. W. Kriwacki, C. J. Oldfield, R. V. Pappu, P. Tompa, V. N. Uversky, P. E. Wright and M. M. Babu. 'Classification of Intrinsically Disordered Regions and Proteins'. In: *Chemical Reviews* 114.13 (July 2014), pp. 6589–6631. ISSN: 0009-2665. DOI: 10.1021/cr400525m.

Vlahoviček, K., L. Kaján, V. Ágoston and S. Pongor. 'The SBASE domain sequence resource, release 12: prediction of protein domain-architecture using support vector machines'. In: *Nucleic acids research* 33.Database issue (Dec. 2004), pp. D223–D225. ISSN: 1362-4962. DOI: 10.1093/nar/gki112. PUBMED: 15608182.

Wilson, D., M. Madera, C. Vogel, C. Chothia and J. Gough. 'The SUPERFAMILY database in 2007: families and functions.' In: *Nucleic acids research* 35.Database issue (Jan. 2007), pp. D308–D313. ISSN: 1362-4962. DOI: 10.1093/nar/gkl910. PUBMED: 17098927.

Wu, C. H., A. Nikolskaya, H. Huang, L.-S. L. Yeh, D. A. Natale, C. R. Vinayaka, Z.-Z. Hu, R. Mazumder, S. Kumar, P. Kourtesis, R. S. Ledley, B. E. Suzek, L. Arminski, Y. Chen, J. Zhang, J. L. Cardenas, S. Chung, J. Castro-Alvear, G. Dinkov and W. C. Barker. 'PIRSF: family classification system at the Protein Information Resource.' In: *Nucleic acids research* 32.Database issue (Jan. 2004), pp. D112–114. ISSN: 1362-4962. DOI: 10.1093/nar/gkh097. PUBMED: 14681371.

2. Identifying high quality profile hidden Markov models

To design a system for crowdsourcing the entries in a protein family database, we should consider how such databases are used. When encountering a protein sequence of unknown function, biologists may query a protein family database, either a specific database or through an aggregator such as Interpro, which queries multiple databases simultaneously (Finn, Attwood *et al.*, 2017). They would also be interested in finding homologues, so they may perform a sequence similarity search. A protein may have the same function as its homologues, and the regions which are conserved across homologues may show where the functional region of the protein is. Sequence similarity search may also be used for identifying the taxa in which the protein is found.

Sequence similarity search engines provided via the Internet such as HMMER (Finn, Clements *et al.*, 2011) and NCBI BLASTP (Altschul *et al.*, 1997) perform a protein family database query and provide protein family matches for the query sequence along with the sequence similarity search results. By integrating these two functions, a sequence similarity search engine provides a comprehensive service for a biologist researching a protein sequence of unknown function.

If the protein sequence in question does not match any models in protein family databases, they may still identify homologues of the sequence using a sequence similarity search. The information encoded in this search represents a gap in the protein family database's knowledge.

In addition to sequence similarity searches which find entirely novel groupings of proteins, it can occur that a search matches all members of an existing family along with further proteins which are yet unclassified in the protein family database. If these proteins are truly homologous with the existing members of the family, then they ought to be members of the family. Therefore, the search encodes a superior model for the existing family.

The novel groupings of proteins identified by the community of sequence similarity search users represent only part of the story for the potential crowdsourced database. As noted in chapter 1, the value of a curated protein family database is found in its annotations. I hypothesise that aside from finding new families, the community is also an untapped resource for

annotating families: A user who has identified a novel grouping through a sequence search is the ideal person to annotate that domain.

In order to allow users to submit groupings of proteins found by sequence similarity search as potential new families, we require the ability to algorithmically identify which searches are sufficiently promising. There are several aspects to determining this, including the degree to which the model represents a true homology, whether the model duplicates a family already in the database (and if so, is the model superior to the family in the database), and whether the proteins which the model matches are of sufficient number (it may not be worthwhile adding a family of only a dozen proteins to the database).

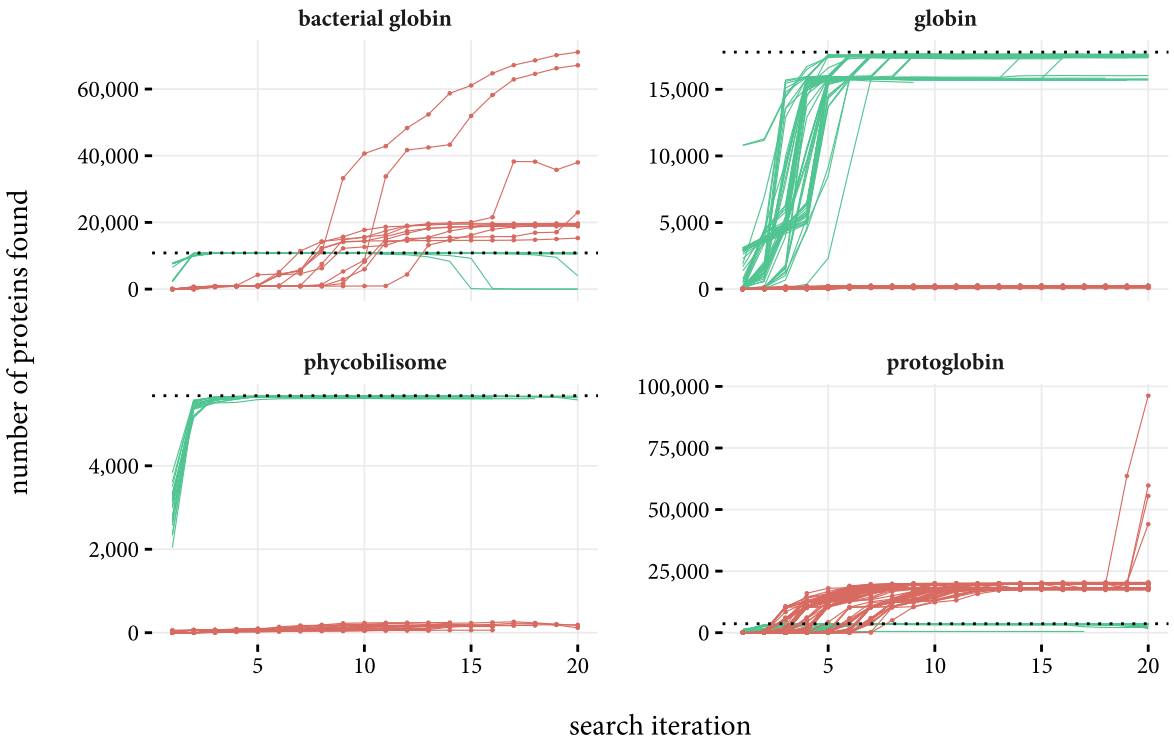
2.1. Quality

Homology between two proteins cannot be directly observed, but can be inferred from comparison of their sequences. Hidden Markov models are a statistical method for performing such inference. We can define the *quality* of a particular model to be the degree to which it captures a homologous group of proteins, to the exclusion of non-homologous proteins. That is, a high quality model will emit protein sequences from the family which it models with high probability, and sequences which are not in the family with low probability. Identifying high quality models is important to facilitate crowdsourcing since it could act as a filter between user submissions and curators. This would reduce the time spent by curators on rejecting low quality models.

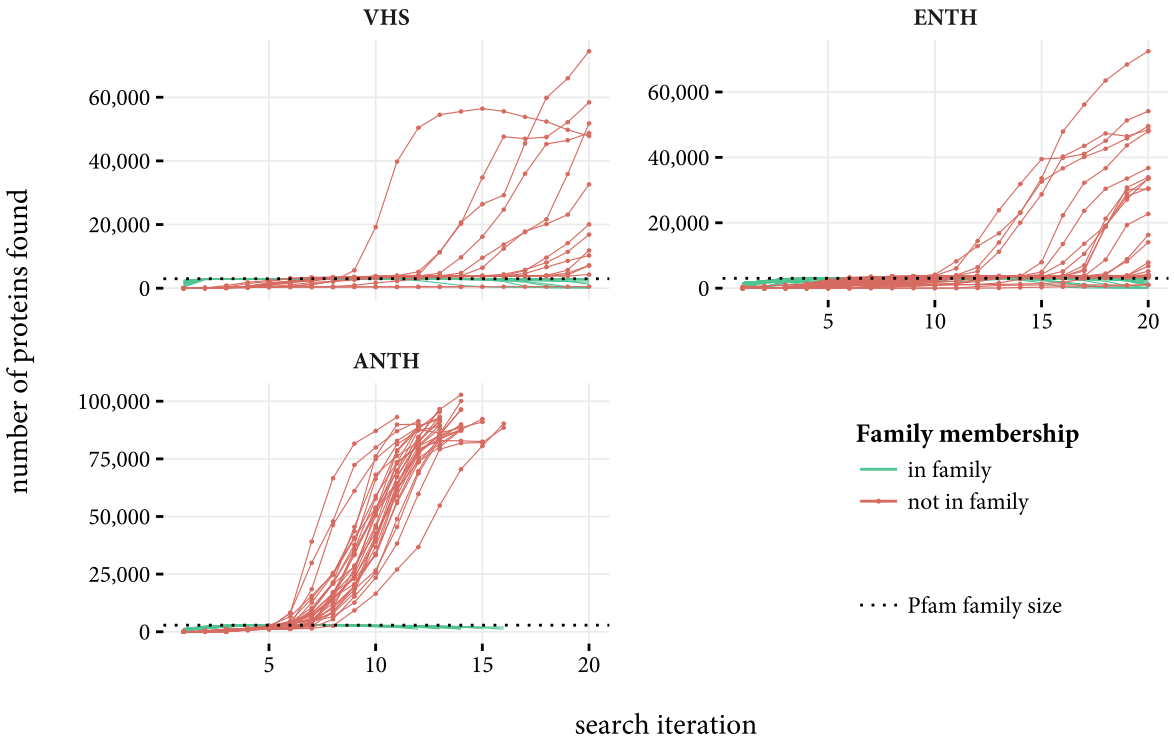
2.1.1. Methods

In order to investigate measurement of quality, it is informative to perform sequence searches which will recapitulate existing Pfam families. Iterative protein sequence similarity searches are often used as a tool for family construction, because of their ability to expand the 'boundaries' of the family by identifying successively more distant homologues (Johnson *et al.*, 2010). Iterative search was contemplated early in the development of protein sequence profile methods for its potential to expand the many sparsely populated families known at the time (Tatusov *et al.*, 1994). This is based upon the concept that with very few members of a family, you have little evidence with which to determine the true distribution of amino acids at each position

[1] This experiment was performed with Pfam release 27.0. This clan was selected because it has been well studied (Lesk and Lecomte, 2013), has numerous protein structures and contains four families, which is neither too small nor excessively large. This release can be retrieved from `ftp://ftp.ebi.ac.uk/pub/databases/Pfam/releases/Pfam27.0/`



a Searches with globin clan query sequences



b Searches with ENTH/ANTH/VHS superfamily clan query sequences

Figure 2.1. For twenty iterations of the *jackhmmer* algorithm, the result set is compared with the Pfam family from which the query sequence was drawn. The number of proteins within the family of the query sequence are shown in green, and the number of proteins not in the family are shown in red. Each line represents the results for one of the query sequences. The size of the family is marked with a horizontal dotted line. For some families, the number of proteins found which are not in the Pfam family are very low, but for others, these 'excess' proteins can subsume the family proteins, sometimes leading to the original query sequence being 'lost'.

		Number of proteins	
Clan	Family	Seeds	Members
Globin (CL0090)	Globin (PF00042)	73	6,000
	Phycobilisome (PF00502)	22	4,056
	Bacterial globin (PF01152)	12	2,140
	Protoglobin (PF11563)	104	1,086
			12,989
EAV (CL0009)	ENTH domain (PF01417)	72	1,127
	ANTH domain (PF07651)	29	1,195
	VHS domain (PF00790)	15	1,706
			3,963

Table 2.1. The number of seed and member proteins in the globin and EAV clan families, and the total number of proteins in the clans, in Pfam release 27.0. Note that the total clan size is less than the sum of the sizes of member families. This is because a protein can be a member of multiple protein families within the same clan.

in the alignment of the sequences. As discussed in section 1.2.2, as proteins are added to the alignment, you can become more certain of what the true distribution of amino acids is, and may therefore be able to identify further members of the family.

I performed *jackhmmer* searches for each of the seed sequences for the globin clan (CL0090)[1]. Summary information for this clan is shown in table 2.1. These searches were limited to 20 iterations. For each search, at each iteration, I compared the result set of proteins with matching regions to the Pfam family from which the seed sequences were drawn by computing the size of the intersection between these two sets, and the number of results which were not present in the Pfam family, which I have termed the excess. In a search which matches the Pfam family closely, we will observe an intersection close in size to the Pfam family and with a small excess. The evolution of these two metrics across iterations of *jackhmmer* is shown in figure 2.1a.

We can observe in the lower left panel of figure 2.1a that each search with a phycobilisome (PF00502) query sequence resulted in a very close match to the Pfam phycobilisome family, as the intersections converge around the size of the Pfam family. It is also the case for globin (PF00042) query sequences that the results find most of the Pfam globin family, with little excess. However, for queries with sequences from the bacterial globins (PF01152) and protoglobins (PF11563), the result sets are much larger than the Pfam families. In addition, we can observe in the upper left panel of figure 2.1a that in several cases, the intersection with the Pfam family actually starts to drop to zero. Once a sequence homology search loses sequences which

we know to be homologous with the query sequence then we should consider the search to be low quality. While experimenting running *jackhmmer* searches on other sequences, I identified another clan which produced many such low quality searches, the ENTH/ANTH/VHS (EAV) superfamily clan (CL0009), and repeated the procedure with the seed sequences for its families. The evolution of the intersection and excess for these searches is shown in figure 2.1b.

For the EAV superfamily, we can observe that many more of the searches rapidly expand in size. We can see the same effect as in the globin clan, where the intersection with the Pfam family starts to drop off, indicating that the search is moving away from the area of protein space occupied by the Pfam family. The fact that the increase in size is followed by the loss of sequences in the intersection shows that the new proteins found by these models may not be homologous with the query sequence, as their addition causes the model to move away from the area of protein sequence space occupied by the query sequence's protein family.

Metrics

As noted, in some cases, the intersection of the search results with the Pfam family from which the query sequence was drawn starts to drop off. When iterative search processes were first introduced, it was recognised that while they could find more distant homologues, the possibility of incorrect homologue detection was also increased. Park *et al.* (1998) investigated three early iterative search methods over essentially two iterations: The initial search using the query sequence, and one additional search based upon the results of the query. Their testing set of protein sequences consisted of pairs of proteins which had been assigned the same superfamily within the structural classification of proteins (SCOP) database, but which had less than 40% sequence identity, in other words, pairs which share homology but which are not 'straightforward' to identify. Their results showed that a profile HMM based method, SAM-T98 outperformed other methods including PSI-BLAST. However, their method for assessing the quality of each query required knowledge of the true homologue pairs in their sequence database.

More recently, Remmert *et al.* (2012) analysed three iterative protein sequence similarity search methods: PSI-BLAST, HMMER and their method, HHblits. They used a similar method to Park *et al.* (1998) to analyse performance, choosing sequences from SCOP superfamilies with less than 30% sequence identity as a test set. In this case, they also compared performance over one to four iterations of each algorithm, finding that further iterations find more homologues, but are also more likely to incorrectly identify non-homologue pairs.

However, for our purposes, we would like to be able to test the quality of searches which find proteins which are not already classified into any Pfam family. Testing the recapitulation of

already known relationships is effective for benchmarking protein sequence similarity search algorithms, but not for the analysis of potential new families. As noted by Eddy (2011), identifying a test sequences for methods in sequence homology search is inherently difficult.

> If we are trying to find methods that detect previously undetectable homologies, no source of real biological sequences will ever be reliably known to be nonhomologous to the benchmark, and we certainly do not want to penalize a powerful method that identifies new true relationships that are currently annotated as nonhomologous 'false positives'.

Still, we have at least one sequence which we know ought to be homologous with all of the search results: The original query sequence. An indicator of a poor quality model would be if it is lost from the search results. That is, the profile HMM generated by the search no longer has a significant match on the query sequence. Clearly, such a profile HMM is no longer a model for the homology of any region of the query sequence. In fact, this occurs in a number of cases for the queries from the EAV superfamily (see figure 2.2). However, the query sequence is only lost some rounds after the search appears to have gone astray, making this metric less useful.

Rather than looking only at the seed sequence, we can look for the presence of all the sequences found in the first iteration of the *jackhmmer* search. These sequences are identified by *phmmer* and will be the most similar to the query sequence. We would expect that a large reduction in first round matches would indicate that the area of protein space which is matched by the model matches is moving, and this could be followed by loss of the query sequence. This situation is called *profile wander* (Bateman and Birney, 2000). Shown in figure 2.3 are the number of first iteration matches for searches with EAV seeds are query sequences. For the 14 result sets with query sequences from the ENTH domain family (PF01417) which contain over 10,000 proteins, half have less than 10% of their first iteration matches remaining. However, this metric also does not show any difference between 'good' and 'bad' searches until later in the search's degeneration than would be useful.

Both of the above metrics attempt to measure quality by the absence of proteins from the result set which ought to be included. That is, proteins for which their exclusion is a false negative. The previous analyses show that these false negatives occur after the result set has grown greatly in size beyond the size of the corresponding Pfam family. This occurs because the large number of newly added proteins has changed the model for the next iteration such that the model no longer matches the query protein or the proteins most similar to the query protein, the first iteration matches. Intuitively, some of these new proteins must be false positives. Since the addition of these false positives is the cause of the false negatives we observe, identifying

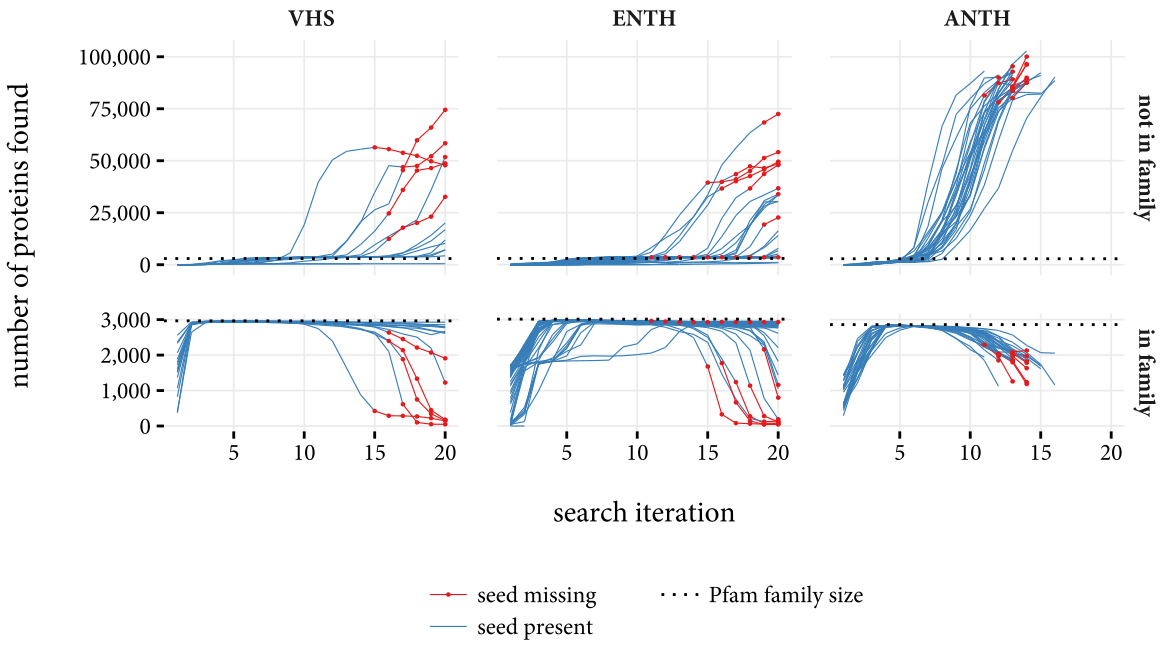


Figure 2.2. The number of proteins in the result set for *jackhmmer* searches with the seeds of the EAV superfamily as query sequences, through twenty iterations. The plot is split between proteins in the result set which are within the family from which the query sequence was drawn, and those without. Rounds in which the query sequence has been lost from the result set are highlighted in red.

when they are added would allow quicker detection of a low quality search. In order to identify false negatives, we could use Pfam family assignments as ground truth. There are major flaws in this approach: *(i)* If the search has truly found a new family which is not present in Pfam, there is no ground truth available; *(ii)* If the search has found a superior model for the search, it will not match proteins which are matched by the Pfam model, which will be scored as false negatives.

Decoy Search

While it is clearly not possible to identify, for an arbitrary model, false negatives, it is possible to identify when the model has matched a false positive, by the use of *decoy* protein sequences, that is, sequences which do not exist, but which 'look' like real protein sequences. A perfect model should not match any of these sequences, since they are not homologous with any real protein.

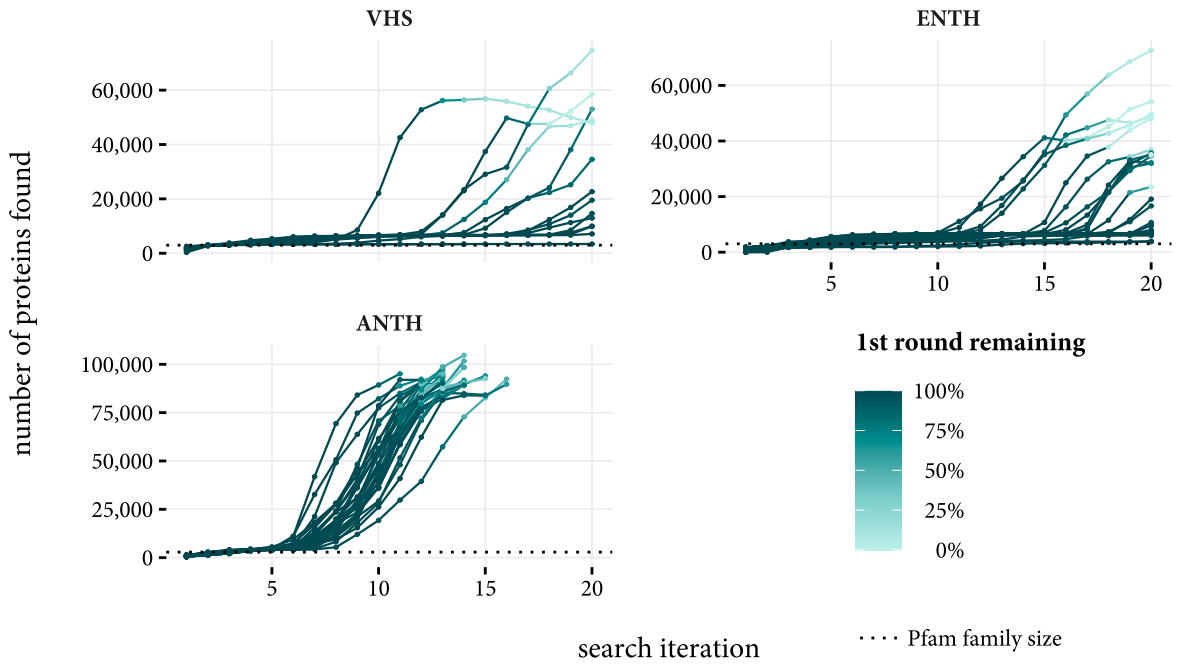


Figure 2.3. The number of proteins in the result set for *jackhmmer* searches with the seeds of the EAV superfamily as query sequences, through twenty iterations, with the percentage of the result set remaining from the first iteration shown.

Decoy sequences are fundamental to both HMMER and SAM, as they are essential aspects of the process by which they score search results. In common with other sequence similarity search methods, they score search results by estimating the probability that the search result is not homologous to the query sequence, and that any similarity is due to chance (Eddy and Wheeler, 2015; Karplus, Barrett *et al.*, 1998; Madera and Gough, 2002). The *bit score* reported by HMMER is the log odds ratio between the probability that the sequence was emitted by the query profile HMM and the probability that the sequence was emitted by the *null model*. In HMMER, the null model emit proteins of random sequence, with composition corresponding to the frequencies of each amino acid in Swiss-Prot[2] (Eddy and Wheeler, 2015). In SAM, the score calculation is formulated identically to HMMER (Barrett *et al.*, 1997). For SAM, the null model is the reverse of the query model (Karplus, Barrett *et al.*, 1998). That is, its transitions are reversed, and the end state becomes the start state. The probability of this reverse null model emitting the query sequence is equal to the probability of the query model emitting the query sequence in reverse.

[2] HMMER actually uses a second null model, for sequences that pass the significance threshold for the first null model, that emits random sequences based on the composition of the query model. This results in more accurate scores for queries with a composition which differs greatly from the average frequency in Swiss-Prot.

In addition to the score, HMMER and other sequence similarity search methods report an *expectation* value or E-value (Eddy and Wheeler, 2015; Madera and Gough, 2002). The E-value for a search result is an estimate of the number of sequences which you would expect to score equal or better than the search result, if the query had been performed against a database of equal size, but containing no sequences homologous to the search query. E-value estimation is based upon extreme value theory, which attempts to characterise the distribution of unlikely events (Eddy, 2008; Madera and Gough, 2002). The unlikely event in this case being sequence similarity by chance rather than by homology. E-value estimation requires the specification of some parameters to the extreme value distribution.

In HMMER, these parameters are *calibrated* by simulation. In HMMER 2, calibration was an optional step performed after model construction, and involved the alignment of (by default) 5000 randomly generated sequences to the model (Eddy, 2001; Madera and Gough, 2002). This step was time consuming, hence it could be omitted, and in this case, conservative estimates were used for the distribution parameters. That is, matches would be, on the whole, reported as less significant than they truly were. From HMMER 3, algorithmic improvements allowed calibration with a less computationally demanding simulation, using three simulations each with 200 randomly generated sequences by default. As such, it is now performed for all models (Eddy, 2008; Eddy and Wheeler, 2015).

In early releases of SAM, the extreme value distribution parameters were not estimated by simulation (Eddy, 2008; Karplus, Karchin *et al.*, 2005). This meant that a time consuming calibration process was avoided. However, due to inaccurate E-value estimates for certain protein structures, model calibration using simulation was later introduced for SAM too.

It may be informative to analyse how models generated by *jackhmmer* behave with decoy proteins. We could produce such a decoy database by generating random sequences. However, Taylor (1986) found that random decoys did not work well as a control, since they are insufficiently similar to real protein sequences. Instead, he introduced the use of reversed protein sequences as decoys, which was subsequently used by SAM (Karplus, Karchin *et al.*, 2005).

I produced two decoy databases based upon pfamseq (the sequence database which Pfam was built prior to version 29.0 (Finn, Coggill *et al.*, 2015)): A reversed database and a random database, generated by shuffling the residues in each sequence in the database. That is, each sequence in the random database directly corresponds to a sequence in the true database, and retains its length and the number of residues of each amino acid. I then queried the databases with the models produced in each round of the *jackhmmer* search of EAV superfamily seeds. The results of this experiment are summarised in figure 2.4. The models do not identify any matches in the reverse database, until the rounds at which we observe the rapid growth in result

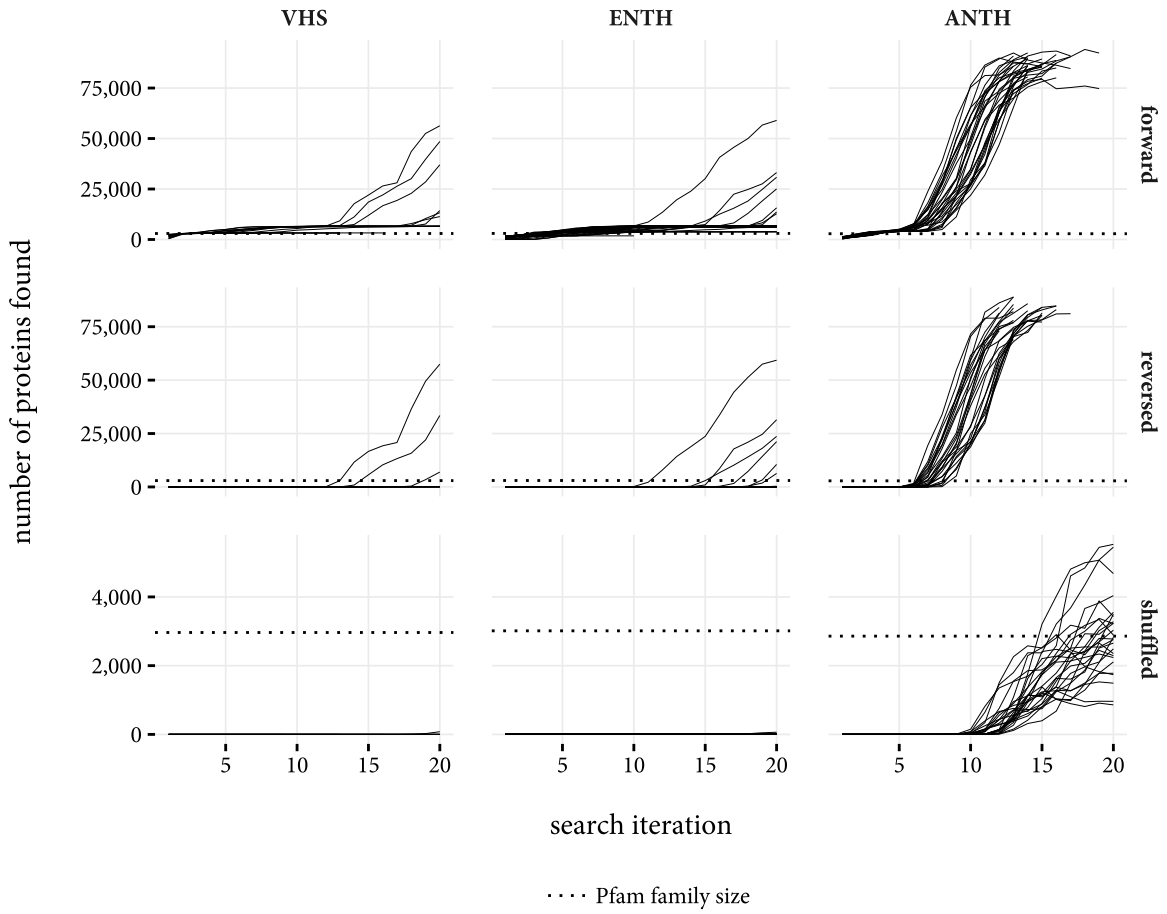


Figure 2.4. The number of proteins in the result set for *jackhmmer* searches with the seeds of the EAV superfamily as query sequences, through twenty iterations, and the number of proteins in the result sets for queries of the models produced by the *jackhmmer* search against the reversed and shuffled decoy sequence databases.

set size, for all three families. The models also do not identify matches in the shuffled decoy database, but remain at zero hits for all twenty iterations for the VHS and ENTH seeds. For the ANTH seeds, the search only gains matches some rounds after the reversed decoy database does. This appears to confirm the observation of Taylor (1986), that a reversed decoy database is more effective than shuffled as a control.

2.2. Set Relationships

Supposing that we wish to assess a potential new family, derived from a user search, for addition to Pfam, we can imagine three scenarios:

1. The search alignment does not overlap with any existing families. It is entirely new residue coverage.
2. The search alignment overlaps with only one existing family.
3. The search alignment overlaps with multiple existing families.

The first case is the clearest case where we would wish to add the family to Pfam. But in the other two cases we may also wish to. In the second case, the search may offer increased residue coverage compared to the existing family. That is, it identifies more members of the family. In the third case, if the multiple families are all of the same Pfam clan, the search may either be a superior model for an existing clan member, or it could be a novel member of the clan. In the second and third case, we require some way of relating an arbitrary search to the families which already exist in Pfam. Since this assessment must be made interactively, as the user performs their search, it needs to be fast. This requirement is discussed in section 2.4.

2.2.1. Set Similarity

The Jaccard index of a pair of sets is a measure of their similarity. It is calculated as follows.

$$J_I(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.1}$$

In words, this is the fraction of all members of A and B which are found in both A and B. Since we are interested in finding search result sets which are supersets of Pfam families, we might also ask what fraction of A is found in both A and B. That is, how close is A to being a subset of B. This measure is known as containment. We can calculate the Jaccard containment as follows (Agrawal *et al.*, 2010; Broder, 1997).

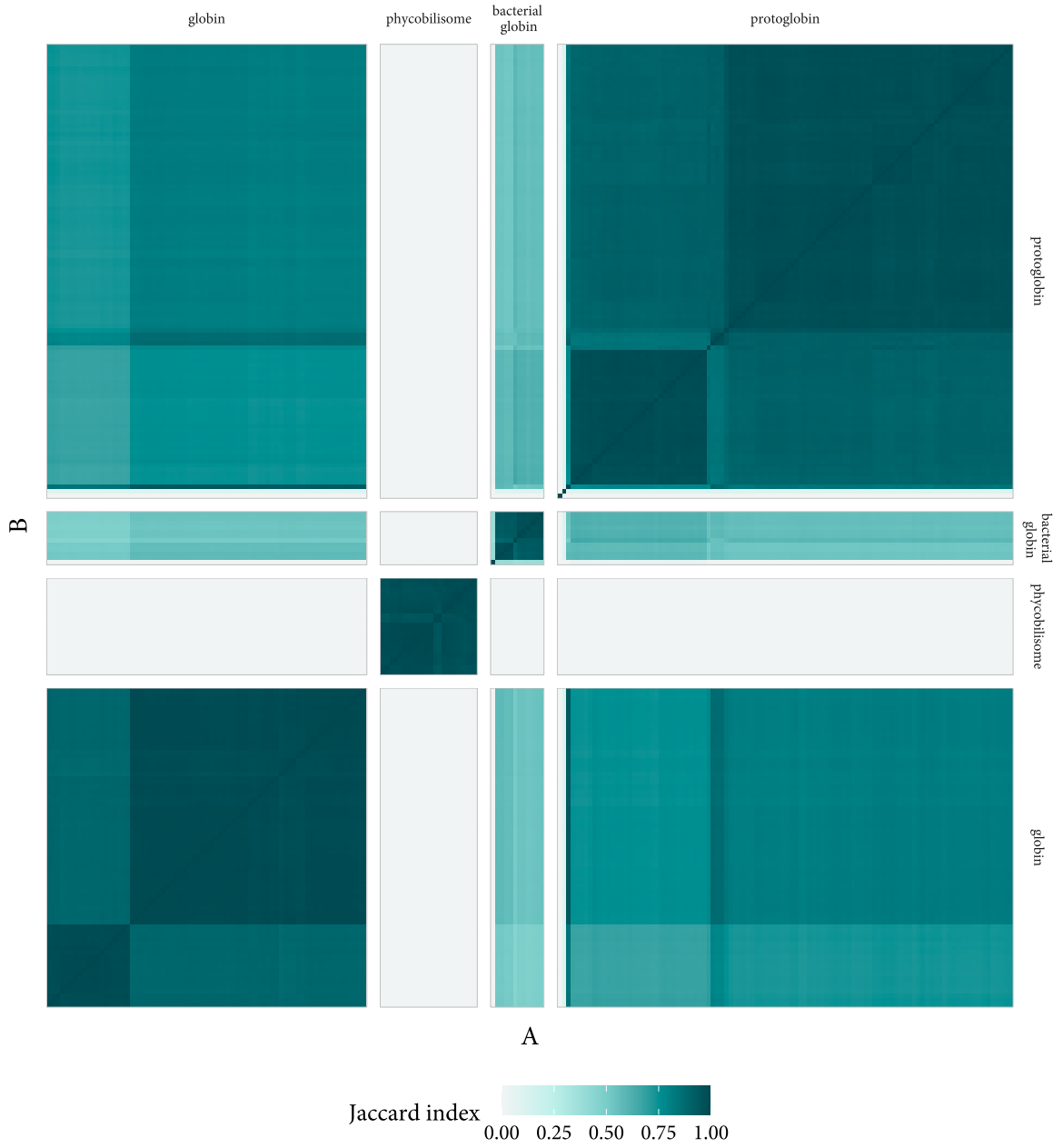


Figure 2.5. Heat map showing Jaccard index between the result sets of the final iteration of *jackhmmer* searches with globin clan seeds as query sequences. Since Jaccard index is commutative, the plot is symmetric in the diagonal. Searches were run until convergence, or the 20th iteration, whichever came sooner.

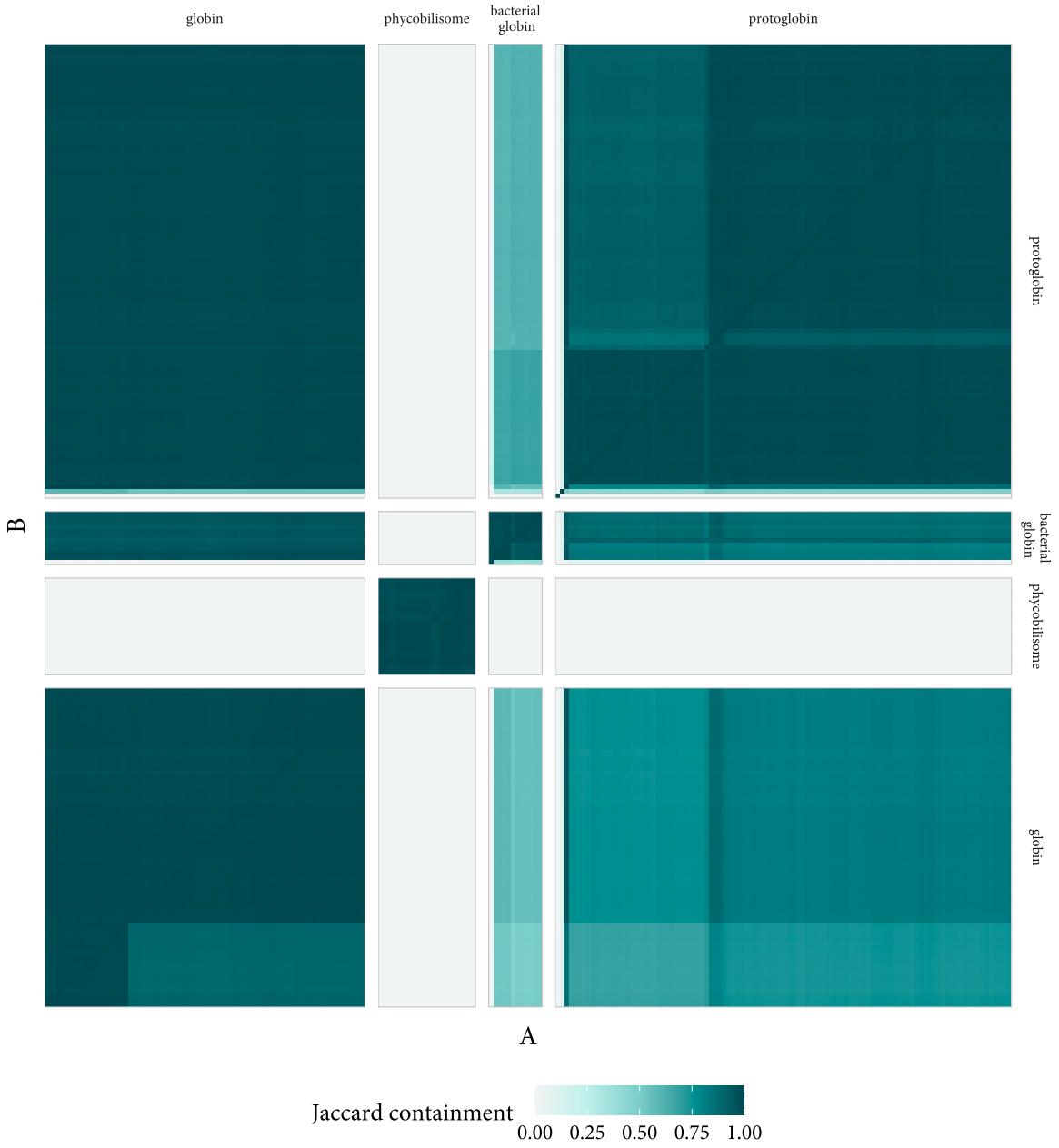


Figure 2.6. Heat map showing Jaccard containment between the result sets of the final iteration of *jackhmmer* searches with globin clan seeds as query sequences. The Jaccard containment is calculated as in equation (2.2). Hence, each value in the plot area scores the degree to which the search with the protein on the x axis is contained by the the search with the protein on the y axis. Searches were run until convergence, or the 20th iteration, whichever came sooner.

$$J_C(A, B) = \frac{|A \cap B|}{|A|} \tag{2.2}$$

To contrast these two metrics, for the *jackhmmer* searches with the seeds of the globin clan (CL0090) relayed in section 2.1, I have calculated the Jaccard index and containment between the result sets of every pair of final iteration models. The Jaccard index values are shown in figure 2.5, and the Jaccard containment in figure 2.6. The 'column' on the left of the plot shows a high Jaccard containment between globin, bacterial globin and protoglobin queries and globin queries. That is, searches with the globin seeds on the x axis are mostly contained by queries with globins, bacterial globins and protoglobin seeds on the y axis. The globin family area of protein space can be found by queries using proteins from any of these families.

2.3. User Searches

In order for this crowdsourcing method to be viable, it must be determined whether user searches would discover new areas of coverage at a significant rate. We would wish to know how often it occurs that a user search alignment falls into the scenarios discussed section 2.2.

User queries and parameters to the HMMER web service are recorded. I retrieved 226 *jackhmmer* queries, and regenerated their search results. Of these, 33 were excluded from further analysis either because the search failed, or because the search query was against the PDB database rather than the UniProt database, leaving a total of 193 searches. I then compared the result sets of the final iteration of the *jackhmmer* query to Pfam's protein family models, to determine whether they overlapped. A search and a Pfam family are only counted as overlapping if the match regions for the two profile HMMs overlap on at least one protein. The results are shown in table 2.2 and figure 2.7 [3].

[3] This experiment was performed with Pfam release 28.0. This release can be retrieved from `ftp://ftp.ebi.ac.uk/pub/databases/Pfam/releases/Pfam28.0/`

We have considered the privacy of HMMER's users in conducting this analysis and the confidentiality of their search results. We will not be closely analysing the individual searches. Where we believe a user has found a novel grouping of proteins, or an improvement over Pfam's, we will not incorporate this into Pfam.

Category	Count	
No overlap	42	22%
Overlaps one Pfam family	38	20%
Overlaps several Pfam families	113	58%
	193	

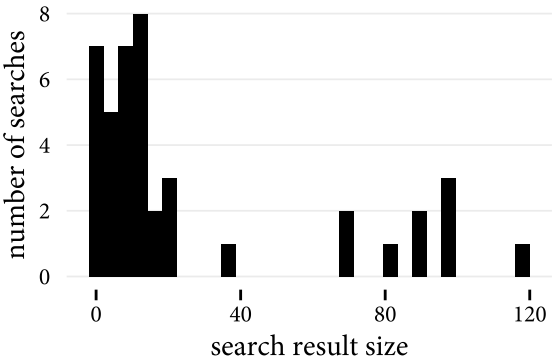
Table 2.2. Analysis of *jackhmmer* user searches, comparing them to existing families in Pfam release 28.0.

Around 40% of searches do not overlap multiple families. Half of these are entirely novel groupings. The bulk of the novel groupings have matches for 20 regions or fewer, but a significant minority match 100 regions or more (see figure 2.7a). The other half overlap with a single family. At most four appear to be potential improvements to an existing family. That is, they contain almost all the members of an existing family, while matching a number of proteins which are not in the family. For example, in figure 2.7b, the upper-rightmost point represents a search which contained over 95% of the members of an existing Pfam family, while adding over twice as many additional proteins to this grouping. As discussed previously, it could be the case that searches which overlap families all within the same clan are potential improvements to the clan.

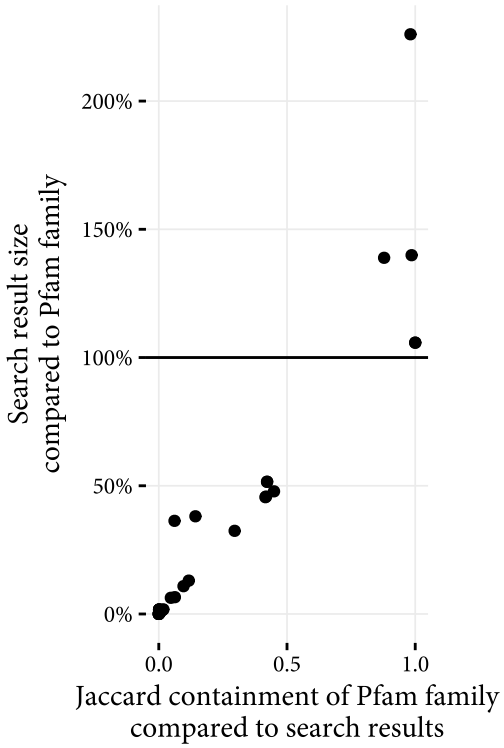
2.4. Fast Set Comparison

In the previous section, I described methods for comparing potential families to existing families to assess their suitability to be added to Pfam. Since we wish to be able to determine whether a user's search is a viable addition to Pfam interactively at the same time that the search results are returned to them, it is necessary to make this decision quickly, ideally taking around a second so as not to negatively impact user experience. However, these methods required calculating the size of the set intersection between the potential new family and the existing families in Pfam.

With a set implementation for which membership is an $O(1)$ operation, the intersection of sets A and B will be $O(\min(|A|, |B|))$. Thus, when comparing a potential new family to all of Pfam, the operation will be bounded above by the number of proteins with matches in the new



a Size distribution of searches which have no overlaps with existing Pfam family.



b Comparison of searches which overlap a single Pfam family to the family they overlap. The Jaccard containment of the family compared to the search is shown on the *x* axis and the size of the search compared to the family is shown on the *y* axis.

Figure 2.7.

family, $|A|$, and by the number of families already in Pfam, n_f, and can therefore be computed in $O(n_f|A|)$ time.

2.4.1. Locality Sensitive Hashing

A hash function transforms variable length data into a fixed length representation (Cormen *et al.*, 2009). That is, given a universe U of possible inputs, a hash function h is a mapping:

$$h : U \to \{1, 2, \ldots, n\} \tag{2.3}$$

where n is the number of possible values of the hash function. Naturally, given the same inputs, a hash function should return the same output. A *collision* occurs when two different inputs produce the same output. Given the unbounded size of U and the fixed size of n, collisions are inevitable, but for most applications it's desirable for them to be rare.

Many applications of hash functions exploit the fact that the output of a hash function is of fixed size. By transforming an object of unbounded size (like a protein family) into a fixed length representation, we can guarantee the length of time that operations upon the object's representation will take.

Locality sensitive hashing is a technique for quickly identifying similar objects. In contrast to a conventional hashing scheme, an effective locality sensitive hashing algorithm will produce collisions when similar objects are input.

MinHash

MinHash is a locality sensitive hash algorithm which estimates the Jaccard index for a pair of sets. Specifically, it estimates the set intersection and union. This additionally allows us to estimate the Jaccard containment. It was was introduced by Broder (1997), with the original application being the elimination of identical web pages from the index of the Alta Vista search engine. MinHash and the Jaccard index and containment for sets A and B are estimated as follows.

For set S, define $\mathrm{MIN}_n(S)$ as

$$\mathrm{MIN}_n(S) = \begin{cases} \text{the } n \text{ smallest elements in } S & \text{if } |S| \geq n; \\ S & \text{otherwise.} \end{cases} \tag{2.4}$$

Let $h(x)$ be some hash function. Define set A' as

$$A' = \{h(x)|x \in A\} \tag{2.5}$$

and B' analogously.

We can then see that

$$\mathrm{MIN}_n(\mathrm{MIN}_n(A') \cup \mathrm{MIN}_n(B')) = \mathrm{MIN}_n(A' \cup B') \tag{2.6}$$

is a sample of at most n elements from $A' \cup B'$, and that

$$\mathrm{MIN}_n(\mathrm{MIN}_n(A') \cup \mathrm{MIN}_n(B')) \cap \mathrm{MIN}_n(A') \cap \mathrm{MIN}_n(B') \tag{2.7}$$

is a sample of at most n elements from $A' \cap B'$ which are also contained in the sample from $A' \cup B'$. We have ensured that these samples are random by hashing the elements of A and B. Hence,

$$\frac{|\mathrm{MIN}_n(\mathrm{MIN}_n(A') \cup \mathrm{MIN}_n(B')) \cap \mathrm{MIN}_n(A') \cap \mathrm{MIN}_n(B')|}{|\mathrm{MIN}_n(\mathrm{MIN}_n(A') \cup \mathrm{MIN}_n(B'))|} \tag{2.8}$$

$$= \frac{|\mathrm{MIN}_n(A' \cup B') \cap \mathrm{MIN}_n(A') \cap \mathrm{MIN}_n(B')|}{|\mathrm{MIN}_n(A' \cup B')|} \tag{2.9}$$

is an estimate of the Jaccard index. To compute this estimate, only the smallest n hashed elements of the sets to be compared is required. A cartoon of this procedure as applied to protein sequence similarity search results is show in figure 2.8.

We can take a similar approach for estimating the Jaccard containment.

We can find

$$\mathrm{MIN}_n(A') \cap B' \tag{2.10}$$

as elements from B' which are also contained in the sample from A' and hence

$$\frac{|\mathrm{MIN}_n(A') \cap B'|}{|\mathrm{MIN}_n(A')|} \tag{2.11}$$

Estimation of Jaccard index with MinHash

1. The user search is represented as a set of protein accessions.
2. The accessions are hashed (in this case, using CRC32).
3. The hashed accessions are sorted. The smallest n hashed accessions form the search's hash.

Hashes for Pfam families are precomputed. By comparing the family's hash with the search's hash, we can obtain an estimate for the Jaccard index between the two.

	Step 1	Step 2	Step 3
User search			
1	W4XLX6	1040815750	10467582
2	W4YI83	2515119136	12749037
3	A0A0D9Z4F0	3460778803	25248169
4	A0A0D3FFX0	487340116	33165958
5	A0A0N4WFH8	2718657979	39770510
6	K0T1L2	2158077265	48572676
7	A0A0E0NS33	1370045315	
8	K9Q0C7	1281340773	
9	U5DJ61	905533730	
10	A0A0C1YNG8	3206662993	
…		…	
4114	H3A6U9	802122903	
Globin (PF00042)			
1	I2H6G3	1020936389	10467582
2	Q2U124	3092201956	25248169
3	A0A0D9RL04	790184029	33165958
4	F1NMZ3	143604913	39770510
5	H3F5M7	3590151977	48572676
6	K0EPI4	3037029836	54787693
7	H2MRF9	1984485617	
8	P62742	2614527290	
9	P02008	2506115589	
10	W5LAJ0	1163737419	
…		…	
17947	A8PGB6	1471850452	
α/β hydrolase fold (PF00561)			
1	A0A089XCC1	1130427396	413885
2	L5JS08	3448699551	433345
3	B1W4M3	3769414139	575926
4	H2RZS5	1310876863	682736
5	A0A096NY42	2547523614	794310
6	A0A067CY03	3136263732	816423
7	R1D859	2952767637	
8	G0MZ56	1734369112	
9	H6RJ57	2049821698	
10	C6D0E6	1690734692	
…		…	
18167	G8BF70	3239745900	
Protoglobin (PF11563)			
1	A0A068JJF3	582126902	12749037
2	Q24SH7	1012862474	15536102
3	Q3IUQ9	3764365142	30485872
4	Q3IS67	701893507	40706356
5	R8BXI4	845681759	42453581
6	M2T1R1	1634761545	44114837
7	Q6D6J9	963792408	
8	B4RFR5	1826543916	
9	G4SZ55	3140454542	
10	Q5B5X9	554411422	
…		…	
3155	G3XSA2	82362689	

Figure 2.8.

is an estimate of the Jaccard containment. Note that for this calculation, it is not essential to use $\text{MIN}_n(A')$ as the random sample from A', but doing so does allow for estimation of both the Jaccard index and containment from a single hash. Assuming we already have the value of $\text{MIN}_n(A')$, these calculations have $\min(n, |B|)$ as their upper time complexity bound rather than $\min(|A|, |B|)$. Since n is a parameter, we can choose how quickly the estimate should be made, at the expense of accuracy (Broder, 1997).

For Pfam, we wish to determine whether a user's search overlaps with an existing family. This comparison is on the basis of amino acid residues. Hence, the elements of the sets to be compared can be represented uniquely as a combination of a protein's identifier and residue position within the protein sequence. We can compute the hashes for every family in Pfam. When a user search is performed, we can compute its hash, and estimate whether it falls into one of the desirable categories above (overlapping no families, or covering a single family or clan, with increased residue coverage).

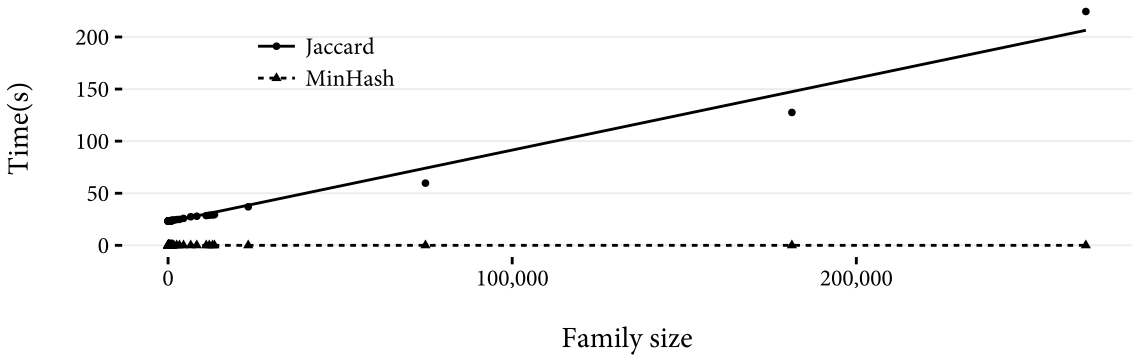


Figure 2.9. Time taken in seconds to calculate the Jaccard index and to estimate the Jaccard index using MinHash, with $n = 800$, between 50 randomly selected Pfam families and every other family in Pfam. A linear least squares best fit line for the two methods is shown.

In order to use this method interactively, the time taken to calculate the hash for the user's search must be taken into account. The hashes for existing Pfam families need only be calculated once, ahead of time, but the hash for a user's search must be calculated while they are waiting for their search results. For a user search with a large number of results, the number of unique residues within protein sequences which the search aligns to could be in the tens of millions. Each of these residues can be thought of as an element of the set, and this set describes the coverage of the search. That is, a user's search result is a list of protein and residue pairs. Concretely, the protein is represented by its UniProt accession, and the residue is represented by a coordinate along the protein sequence. Hence, each search is a set of accession–residue

coordinate pairs, which must be hashed and sorted in order to produce the search's hash. This hash is a compact representation of the entire result set of the search.

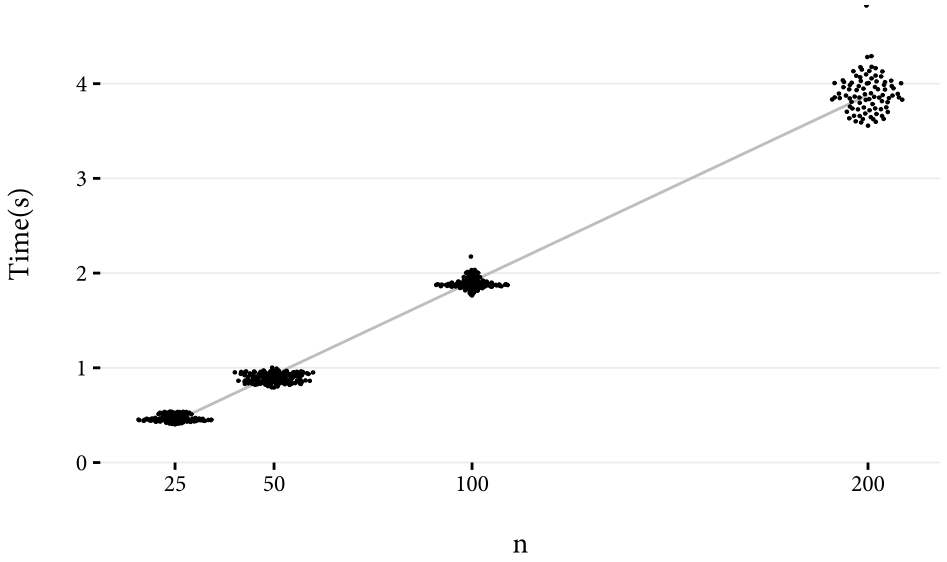
The number of set elements can be reduced by sacrificing accuracy. Ranges of residues of arbitrary size can be grouped together in *chunks*, where the size of the chunk is defined by w, the '*window*' size. If any of the residues in each chunk of the protein sequence is matched by the search, then the assumption is made that all of the residues falling in the chunk are matched by the search. Hence, when residues are chunked together, each search is represented as a set of accession-chunk coordinate pairs, where the presence of a particular chunk in the set means that at least one of the residues falling in the chunk was matched by the search. As the size of these chunks increases, the number of elements is reduced, but the risk that a search which does not overlap with a Pfam family is misidentified as overlapping with the family will increase. The chunk which a residue with coordinate i should be assigned to is computed as $\lfloor i/w \rfloor$.

I implemented MinHash in the Python programming language to validate its theoretical gain in performance over exact calculations. I generated hashes of every family in Pfam 29.0 (Finn, Coggill *et al.*, 2015). I chose 50 random families from Pfam, and for each of these I timed the calculation of the Jaccard index between the family and every family in Pfam, and the MinHash estimate for the Jaccard index with n values of 25, 50, 100, and 200. For each method, the calculation was repeated three times, and the minimum of the three used. The results are shown in figure 2.9.

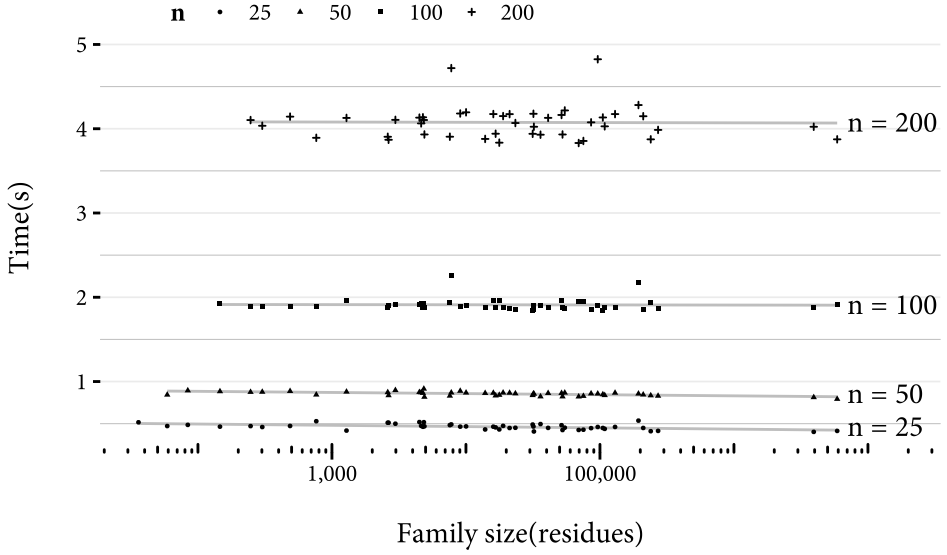
For any family size, MinHash is faster. Also clear is the linear relationship between family size and calculation time for the Jaccard index. In figure 2.10 the linear relationship between n and calculation time for MinHash is shown, and so is the constant time to estimate the Jaccard index as family size varies. In figure 2.11, calculation of the Jaccard containment grows with $\log(n)$. However, for the family sizes tested, calculating the Jaccard containment was faster than the Jaccard index. This is due to the sort operation required to estimate the Jaccard index.

Calculation time of the order of seconds, even with high values of n, will enable fast estimation of the relationship between a potential new family and the rest of Pfam. In figure 2.12, the concordance between Jaccard index and containment and their MinHash estimates, between the same sample as above and the rest of Pfam are shown. Even with $n = 25$ the discrepancy is not great. Thus, searches which may not overlap any existing Pfam family, and families which may be improvements over existing families can be identified in less than a second.

Increasing the value of w also reduces accuracy, but reduces the time to compute the hash of the potential new family. In figure 2.13, the time taken to compute hashes for different values of w is shown. With a w value of 1 (that is, without chunking residues), it takes over 10 seconds to compute the hash of the largest family. Increasing w enables this time to be reduced to under

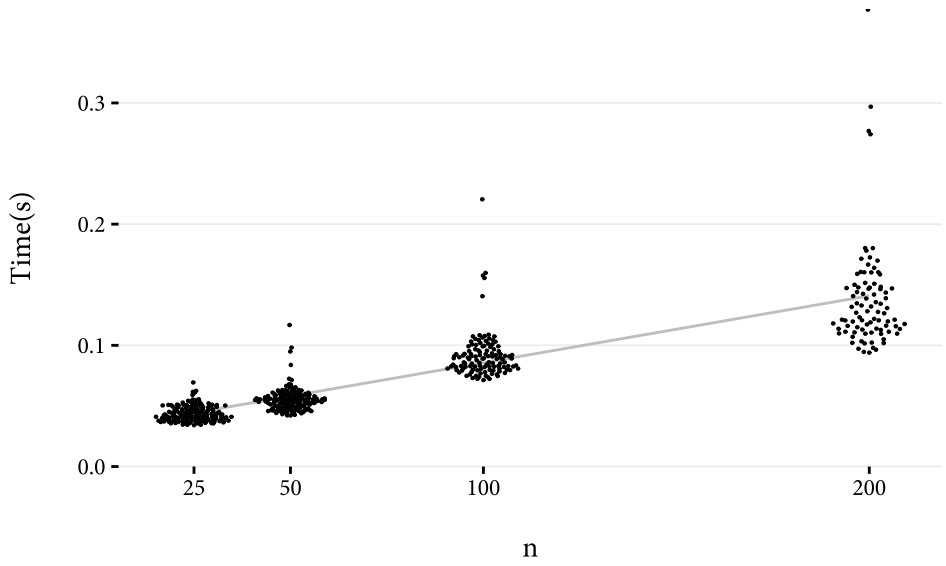


a Time taken in seconds to estimate the Jaccard index using MinHash with n values of 25, 50, 100, and 200, with w value of 1 (that is, no chunking of residues). A linear least squares best fit is shown. Note that the data points are jittered on the x axis to better show their distribution.

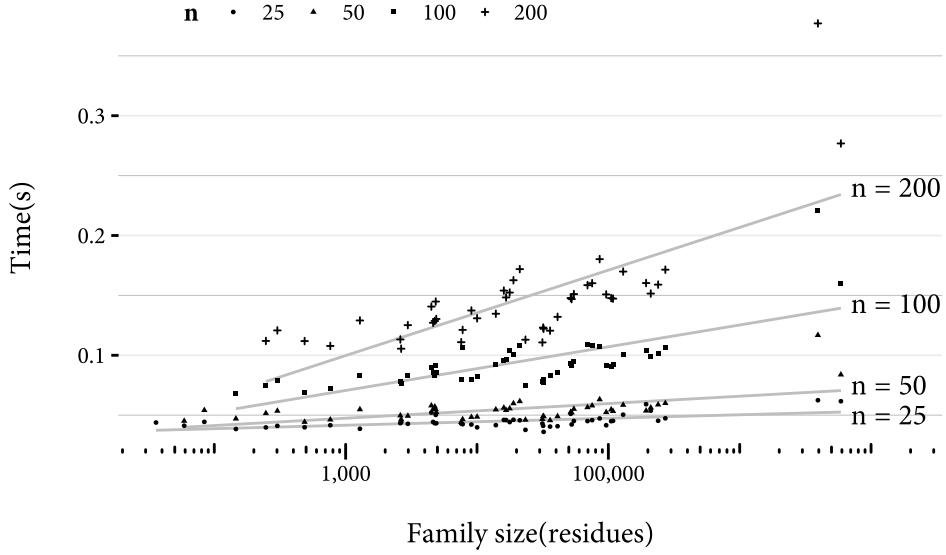


b Time taken in seconds to estimate Jaccard index against family size on a logarithmic scale, with n values of 25, 50, 100, and 200, plotted with circles, triangles, squares and crosses respectively. A linear least squares best fit for each n is shown.

Figure 2.10. Time to estimate Jaccard index by n and family size, for 50 families randomly selected from Pfam 29.0. The time taken for the Jaccard index between the family and the rest of the families to be estimated is shown. For cases where the family size is less than n, the results were excluded from the plot.

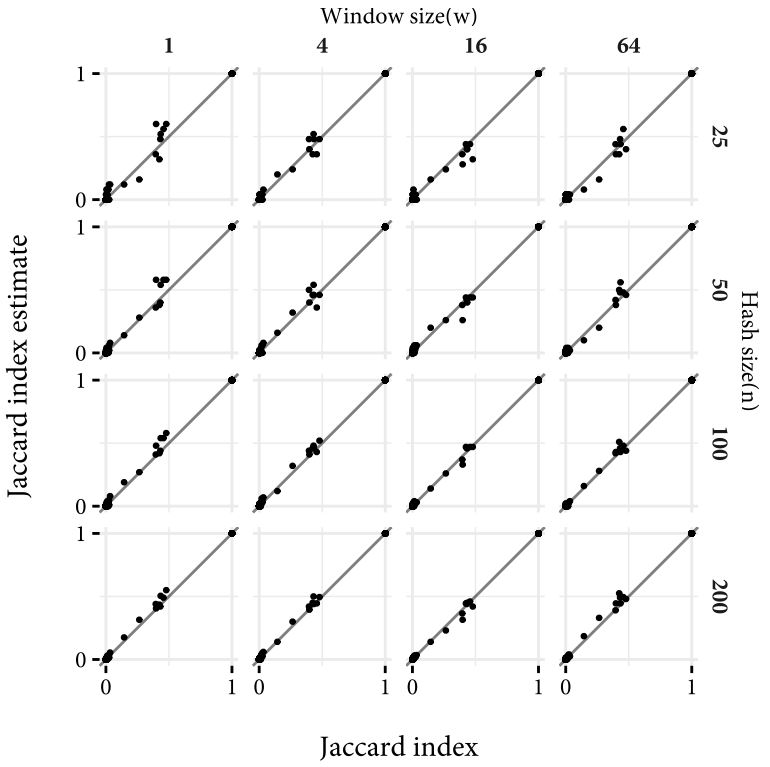


a Time taken in seconds to estimate the Jaccard containment using MinHash with n values of 25, 50, 100, and 200, with w value of 1 (that is, no chunking of residues). A linear least squares best fit is shown. Note that the data points are jittered on the x axis to better show their distribution.

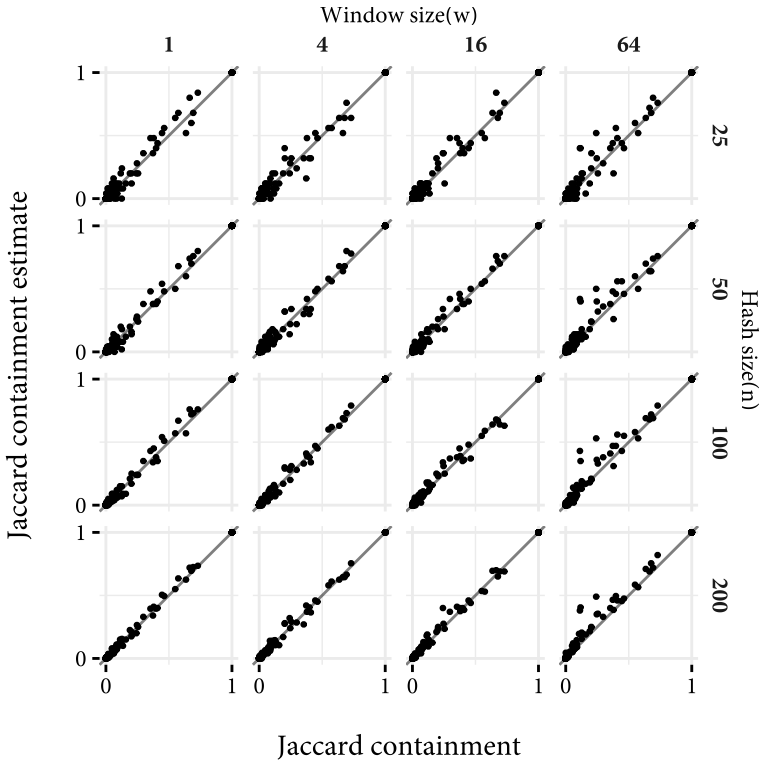


b Time taken in seconds to estimate Jaccard containment against family size on a logarithmic scale, with n values of 25, 50, 100, and 200, plotted with circles, triangles, squares and crosses respectively. A linear least squares best fit for each n is shown.

Figure 2.11. Time to estimate Jaccard containment by n and family size, for 50 families randomly selected from Pfam 29.0. The time taken for the Jaccard index between the family and the rest of the families to be estimated is shown. For cases where the family size is less than n, the results were excluded from the plot.



a Jaccard index concordance.



b Jaccard containment concordance.

Figure 2.12. The Jaccard index and containment, and the MinHash estimate of these values between 50 randomly selected Pfam families and every other family in Pfam, for different values of n and w. The diagonal lines show the position that a perfect estimate would fall.

a second. For a production system, regular waits of over 10 seconds would be unacceptable, so w should be set to at least 4. On the other hand, high values of w will result in more frequent errors in multidomain proteins: In cases where the domains have fewer residues separating them than $w + 1$, there is the possibility that the profiles for the two domains could be wrongly identified as overlapping. Therefore, w of greater than 16 could be detrimental.

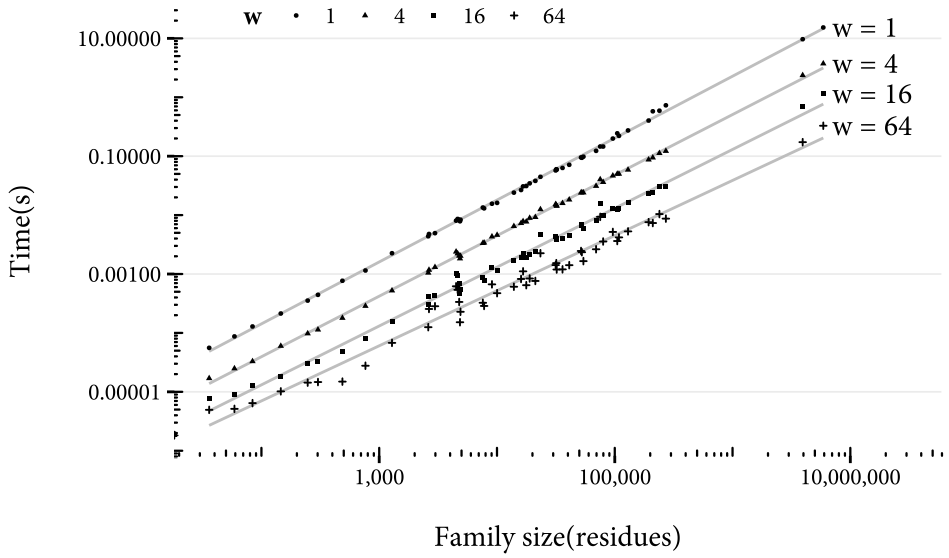


Figure 2.13. Time taken in seconds to compute the MinHash set hash for against family size, both on a logarithmic scale, for w values of 1, 4, 16 and 64.

2.5. Conclusion

In this chapter I have introduced the protein family database as a method of protein sequence analysis. I have described how sequence profiles are used to define these families and in particular, how profile HMMs are exploited to identify homologues.

Following this, I have explored how the HMMER software package can be used to create protein family sequence profile HMMs, and how such models can vary in quality. High quality profile HMMs identify homologous protein sequences. Low quality models may identify many false positives, which do not have shared homology. It is possible to construct profile HMMs which match unclassified sequences which do not share any homology.

By querying the models against a decoy database sequences, I found that low quality models often identified false homology between real sequences and artificial sequences, with reversed sequences being more likely to 'fool' the model. This is perhaps puzzling, since much effort

over the past twenty years has gone into preventing profile HMM sequence similarity search from identifying false homology, and these methods use decoy proteins as their basis (Eddy, 2011; Karplus, Barrett *et al.*, 1998). As previously noted, using reversed protein sequences as decoys is not a new method, so it may be surprising that HMMER is susceptible to this trick.

Mistry *et al.* (2013) performed *hmmsearch* queries with Pfam families, and analysed families which matched regions which also matched other families (that is, overlapping families). They found that families with coiled-coil regions were overrepresented amongst these overlapping families, suggesting that certain common structural motifs could lead to HMMER identifying shared homology when none exists. The authors (among them Sean Eddy) identify that the fundamental cause of this problem is that null models in HMMER, and other sequence similarity search methods, are insufficiently sophisticated. However, conversely, in more sophisticated null models, more problems can be introduced. Eddy (2011) notes that 'reversed sequences are surprisingly significantly more likely to show a significant match to the original sequence (because of a counterintuitive statistical effect of the frequency of approximate palindromes in any sequence)', and as such they are 'problematic as a source of nonhomologous segments in automated benchmarking'.

Reverse sequence decoys may be unsuitable for use internally within HMMER for E-value calibration, but I believe that in this application of protein sequence similarity search to crowd-sourced identification of new or improved families, they may have utility. Since the process as envisioned is *not* automated, matches against reverse decoys could act as more of a warning sign, to be interpreted by curators, rather than an automatic filter.

This method could be used in practice either by querying all HMMER searches against the reverse sequence decoy database in addition to the sequence database, but this would double compute time and storage requirements. Alternatively, a subsample of the reverse sequence database could be used, reducing the compute and storage requirements, but possibly reducing sensitivity. It is also feasible to produce a 'reversed' profile HMM, by reversing the direction of the model's transitions, which could be queried against the forward sequence database, eliminating the need for a reverse database to be created and stored, but with similar compute requirements to querying against the full reverse sequence database.

Finally, I have developed a method for quickly comparing the search results produced by querying a profile HMM against a protein sequence database, to a protein family database. I have adapted a method for estimating the Jaccard index of a pair of sets to estimate the Jaccard containment. This allows the rapid evaluation of the relationship between a pair of multiple sequence alignments. That is, does one alignment contain a superset or subset of the regions in the other.

In the context of the HMMER web service, this method may seem a redundant: When performing a *phmmer* search with the HMMER web service, the application identifies sequences in the search results with common 'architecture', which shows the position of domains along every sequence in the search results. Naively, one might think that in order to construct this, an alignment for every result sequence against every Pfam profile HMM would need to be calculated. In fact, this alignment is pre-calculated for every sequence in UniProt, so the architectures are found by a database query. This same database contains all the information needed to find overlap between protein families, and between a potential new family and the rest of Pfam. So what benefits does the method described in this chapter provide?

The main benefit of this method is that the storage requirements required to perform this sort of analysis is dramatically reduced. The MySQL table storing Pfam matches against UniProt is 45 GiB. In contrast, the storage requirements for the hashes tested in this chapter never exceeded 1 GiB, even with $w = 1$ and $n = 2000$. With $w = 8$ and $n = 100$, a Gzip compressed JSON file containing the Pfam hashes occupies 16.3 MiB. This comparison could therefore be performed locally using JavaScript in a user's web browser, rather than requiring additional database queries. This could be useful in a web application in which users are able to modify the boundaries of an alignment, and see changes in overlaps with existing Pfam families reflected in realtime.

These two methods are intended to enable an automated quality control for user submitted protein family profile HMMs. The MinHash derived comparison method for protein families is a critical component of a automated pipeline for identifying families which are candidates for integration with Pfam. This method can be adjusted to meet the required speed for an interactive protein sequence similarity search by slightly reducing the accuracy of the estimate. This is complemented by the method for identifying low quality models and together these two methods can be used to identify high quality profile HMMs which improve on Pfam's existing classification.

2.6. References

Agrawal, P., A. Arasu and R. Kaushik. 'On indexing error-tolerant set containment'. In: *Proceedings of the 2010 international conference on management of data*. New York, USA: ACM Press, June 2010, p. 927. ISBN: 9781450300322. DOI: 10.1145/1807167.1807267.

Altschul, S. F., T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman. 'Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.' In: *Nucleic acids research* 25.17 (Sept. 1997), pp. 3389–3402. ISSN: 0305-1048. PUBMED: 9254694.

Barrett, C., R. Hughey and K. Karplus. 'Scoring hidden Markov models'. In: *Bioinformatics* 13.2 (Apr. 1997), pp. 191–199. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/13.2.191.

Bateman, A. and E. Birney. 'Searching databases to find protein domain organization'. In: *Analysis of Amino Acid Sequences*. Ed. by P. Bork. Vol. 54. Advances in Protein Chemistry. Elsevier, 2000, pp. 137–157. ISBN: 9780120342549. DOI: 10.1016/S0065-3233(00)54005-4.

Broder, A. 'On the resemblance and containment of documents'. In: *Compression and complexity of sequences*. Selerno, IT: IEEE, 1997, pp. 21–29. ISBN: 0-8186-8132-2. DOI: 10.1109/SEQUEN.1997.666900.

Cormen, T. H., C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*. 3rd ed. MIT Press (MA), 2009, p. 1292. ISBN: 9780262533058. URL: http://books.google.co.uk/books?id=h2xRPgAACAAJ%7B%5C&%7Ddq=isbn:9780262533058%7B%5C&%7Dhl=%7B%5C&%7Dcd=1%7B%5C&%7Dsource=gbs%7B%5C_%7Dapi%20papers3://publication/uuid/2F25E91B-30A0-4B1B-B58C-DF34185EBD59.

Eddy, S. R. 'Accelerated Profile HMM Searches.' In: *PLoS computational biology* 7.10 (Oct. 2011), e1002195. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002195. PUBMED: 22039361.

Eddy, S. R. 'A Probabilistic Model of Local Sequence Alignment That Simplifies Statistical Significance Estimation'. In: *PLoS Computational Biology* 4.5 (May 2008). Ed. by B. Rost, e1000069. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1000069.

– *HMMER User's Guide*. Aug. 2001. URL: eddylab.org/software/hmmer/2.2g/Userguide.pdf.

Eddy, S. R. and T. J. Wheeler. *HMMER User's Guide*. Feb. 2015. URL: eddylab.org/software/hmmer3/3.1b2/Userguide.pdf.

Finn, R. D., T. K. Attwood, P. C. Babbitt, A. Bateman, P. Bork, A. J. Bridge, H.-Y. Chang, Z. Dosztányi, S. El-Gebali, M. Fraser, J. Gough, D. Haft, G. L. Holliday, H. Huang, X. Huang, I. Letunic, R. Lopez, S. Lu, A. Marchler-Bauer, H. Mi, J. Mistry, D. A. Natale, M. Necci, G. Nuka, C. A. Orengo, Y. Park, S. Pesseat, D. Piovesan, S. C. Potter, N. D. Rawlings, N. Redaschi, L. Richardson, C. Rivoire, A. Sangrador-Vegas, C. Sigrist, I. Sillitoe, B. Smithers, S. Squizzato, G. Sutton, N. Thanki, P. D. Thomas, S. C. E. Tosatto, C. H. Wu, I. Xenarios, L.-S. Yeh, S.-Y. Young and A. L. Mitchell. 'InterPro in 2017—beyond protein family and domain annotations'. In: *Nucleic acids research* 45.D1 (Jan. 2017), pp. D190–D199. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1107. PUBMED: 27899635.

Finn, R. D., J. Clements and S. R. Eddy. 'HMMER web server: interactive sequence similarity searching.' In: *Nucleic acids research* 39.Web Server issue (July 2011), W29–W37. ISSN: 1362-4962. DOI: 10.1093/nar/gkr367. PUBMED: 21593126.

Finn, R. D., P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, G. A. Salazar, J. Tate and A. Bateman. 'The Pfam protein families database: towards a more sustainable future.' In: *Nucleic acids research* 44.D1 (Jan. 2015), pp. D279–D285. ISSN: 1362-4962. DOI: 10.1093/nar/gkv1344. PUBMED: 26673716.

Johnson, L. S., S. R. Eddy and E. Portugaly. 'Hidden Markov model speed heuristic and iterative HMM search procedure.' In: *BMC bioinformatics* 11.431 (Jan. 2010). ISSN: 1471-2105. DOI: 10.1186/1471-2105-11-431. PUBMED: 20718988.

Karplus, K., C. Barrett and R. Hughey. 'Hidden Markov models for detecting remote protein homologies'. In: *Bioinformatics* 14.10 (Nov. 1998), pp. 846–856. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/14.10.846.

Karplus, K., R. Karchin, G. Shackelford and R. Hughey. 'Calibrating E-values for hidden Markov models using reverse-sequence null models'. In: *Bioinformatics* 21.22 (Aug. 2005), pp. 4107–4115. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti629. PUBMED: 16123115.

Lesk, A. M. and J. T. Lecomte. 'The Globin Family'. In: *Protein Families*. Ed. by C. Orengo and A. Bateman. Hoboken, NJ, USA: John Wiley & Sons, Inc., Nov. 2013, pp. 207–235. DOI: 10.1002/9781118743089.ch9.

Madera, M. and J. Gough. 'A comparison of profile hidden Markov model procedures for remote homology detection.' In: *Nucleic acids research* 30.19 (Oct. 2002), pp. 4321–4328. ISSN: 1362-4962. PUBMED: 12364612.

Mistry, J., R. D. Finn, S. R. Eddy, A. Bateman and M. Punta. 'Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions'. In: *Nucleic Acids Research* 41.12 (July 2013), e121. ISSN: 1362-4962. DOI: 10.1093/nar/gkt263.

Park, J., K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard and C. Chothia. 'Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods'. In: *Journal of Molecular Biology* 284.4 (Dec. 1998), pp. 1201–1210. ISSN: 0022-2836. DOI: 10.1006/JMBI.1998.2221.

Remmert, M., A. Biegert, A. Hauser and J. Söding. 'HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment'. In: *Nature Methods* 9.2 (Feb. 2012), pp. 173–175. ISSN: 1548-7091. DOI: 10.1038/nmeth.1818.

Tatusov, R. L., S. F. Altschul and E. V. Koonin. 'Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks.' In: *Proceedings of the National Academy of Sciences of the United States of America* 91.25 (Dec. 1994), pp. 12091–12095. ISSN: 0027-8424. PUBMED: 7991589.

Taylor, W. R. 'Identification of protein sequence homology by consensus template alignment'. In: *Journal of molecular biology* 188.2 (Mar. 1986), pp. 233–258. ISSN: 0022-2836. DOI: 10.1016/0022-2836(86)90308-6. PUBMED: 3088284.

3. Identifying literature relevant to family curation

A family in a protein family database is composed of two parts. The first is the alignment of protein sequences and the profile hidden Markov model (HMM) which make up the family, the subject of the previous chapter, and the second is the annotations, which will be discussed in this chapter. Annotations in a protein family database give information about the possible function and structure of family members. Further information which might be annotated to a family include tissue specificity, subcellular localisation or active site positions. For example, the family T3SSipB (PF16535) is described by a curator as follows.

> T3SSipB is a family of pathogenic Gram-negative bacterial proteins that invade human intestinal cells via the type III secretion system translocators. T3SSipB represents the coiled-coil region of the proteins and is shown to be homologous in activity to the pore-forming toxins of other Gram-negative pathogens, such as colicin Ia.

This is followed by a list of the references from which this information has been sourced. The most basic annotation is to assign a name to a family. In Pfam, annotation is performed by curation staff. Curators use the literature and their own expertise to annotate protein families. These annotations and references may give sufficient information that users may not need to perform their own literature searches.

The goal of this thesis is to facilitate the addition of crowdsourced protein families to Pfam. The previous chapter has outlined some methods for the identification of candidate protein families from protein sequence similarity search results. It would also be desirable for protein sequence similarity search users who have identified such candidate protein families to be able to perform some of the annotation work required to construct a Pfam family. This is appealing for two reasons. Firstly, it would save curator time, and second, biologists who identify novel protein families through sequence similarity search may have expert knowledge about the members of the family. One could imagine that the task of staff curators would be to scru-

tinise the quality of submitted families, and modify them for inclusion to Pfam, rather than to perform all of the annotation.

In this chapter I describe work on identifying literature relevant to the curation of family, based on the proteins found in a protein sequence similarity search.

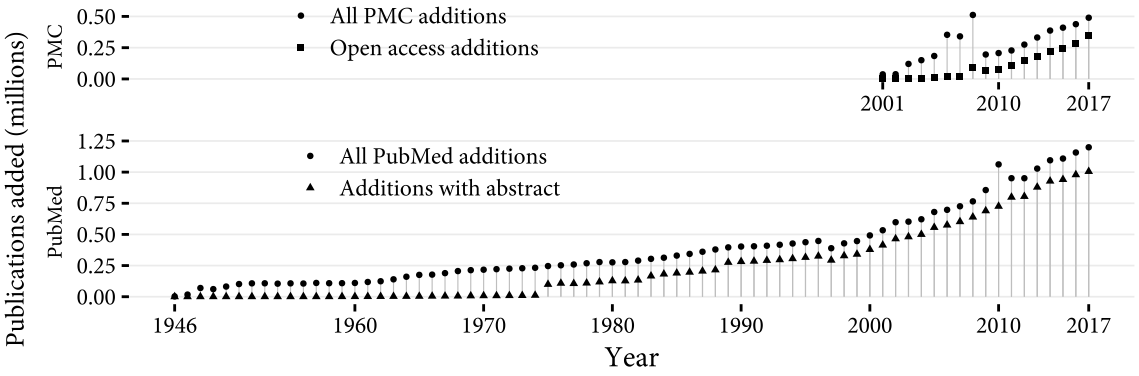
3.1. Background

3.1.1. Literature resources

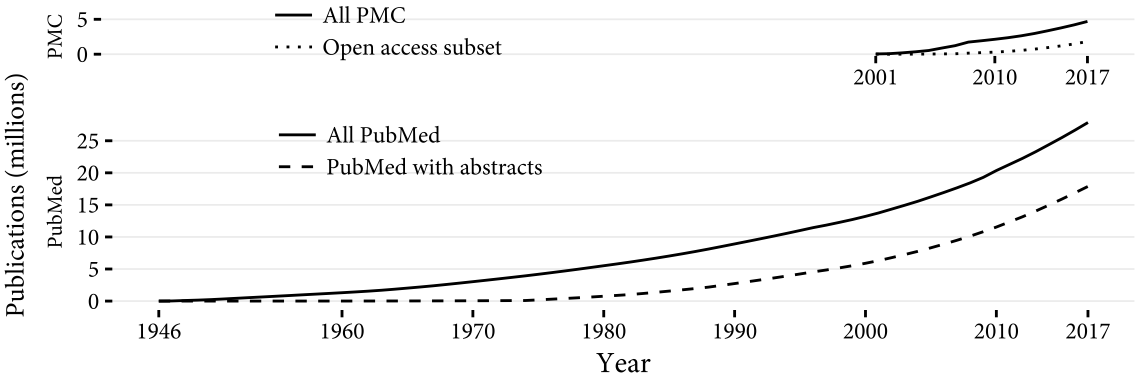
PubMed is an important resource for literature in the life sciences and some other disciplines. PubMed is part of a cluster of resources operated by the United States National Library of Medicine (NLM). PubMed is the name for an abstract database and search engine, which incorporates abstracts from several sources, the most important source being MEDLINE, which is also produced by the NLM (NCBI Resource Coordinators, 2018). MEDLINE has been in operation since 1971, and is a continuation of the NLM's *Index Medicus*, an index to medical publications, which has been produced since 1879. As such, MEDLINE was originally intended as a resource for clinicians and medical researchers (Coletti and Bleich, 2001). The MEDLINE database is a curated resource. In addition to the abstract text, NLM staff annotate abstracts with Medical Subject Headings (MeSH), which are a controlled vocabulary describing the subject of the publication. MeSH terms are only added to abstracts within the life sciences. PubMed also indexes journals from areas outside the life sciences. These are not given MeSH terms, and are not part of MEDLINE.

The NLM also produces a resource called PubMed Central (PMC). PMC is a full-text repository, in contrast to PubMed, which accepts only abstracts. PMC is not a subset of PubMed as it contains some material which is not accepted by PubMed. However, the majority of the publications in PMC have their abstract indexed by PubMed. Both PubMed and PubMed Central are accessible on the web through their respective search engines. The full content of PubMed is available to download in Extensible Markup Language (XML) format. In contrast, only the 'open access subset' of PubMed Central is available for download. Shown in figure 3.1 are the sizes of these resources over time.

Europe PMC is produced by the European Bioinformatics Institute (EMBL-EBI) (Europe PMC Consortium, 2015). It mirrors a full copy of PubMed and PMC, augmented by a patent archive, and abstracts from other sources. In contrast to the NLM's PubMed and PubMed Central, Europe PMC allows the searching of both abstracts and full text simultaneously. That is, one can search both the PubMed and PubMed Central databases at the same time. However, Europe



a Year on year growth of NLM literature resources.



b Total size of NLM literature resources by year.

	Number of publications (millions)	
Resource	Total	Available for text mining
PubMed	28.5	18.4
PubMed Central	5.0	2.1

c The number of publications in the NLM literature resources, and the number available for use in text mining. These data were collected by searching in PubMed and PubMed Central for the articles in their respective collection within the date ranges. Note that this is a retrospective analysis of the publications presently in PubMed, based on their stated date of availability.

Figure 3.1.

PMC depends on the NLM for the production of the majority of its index. Europe PMC also operates the Europe PMC Plus service, which is a full-text depository.

3.1.2. Literature search for curation

Developing tools which use text mining to assist curation of biological databases (biocuration) is an area of active research. It's important to consider that literature curation for a biological database can occur in two 'directions'. The curator could start from the literature. That is, they have identified a publication which discusses a biological entity, such as a gene or protein, which would be desirable to include in the database. After confirming that it isn't already included in the database, they may gather further literature evidence, and then construct a new entry. Alternatively, the curator may start with an entity which they wish to include, and then try to identify literature about the entity.

The UniProt curators use literature search tools to identify curatable literature (that is, literature which describes a protein which could be integrated into UniProt), and then integrate the information from the literature into existing entries or into new entries (Poux *et al.*, 2017). Identifying curatable literature is referred to as document triage. Triage for UniProt uses the text mining tool PubTator to search for literature which mentions proteins, and then prioritises the results based on the frequency of mentions of proteins.

In contrast, the task which we are interested in performing is identification of literature relevant to a newly constructed protein family. Supposing that a user has identified a family which may be incorporated into Pfam, we would like to present them with articles which are most likely to contain information relevant to the family. In the case of a Pfam family, the literature which is curated into the entry is often not about a protein family in general, but about a small subset of its member proteins. The curator may integrate information from multiple sources to build up a picture of the function or structure of a protein family. Many members of the family may not have been studied in detail before, so the curator must use the protein family alignment along with literature about its members to deduce what information is applicable to the family as a whole. Literature describing the 3D structure of a protein may be especially useful, since such papers may summarise information about a whole family of proteins, and the structure may help in identifying homologues. Some protein families have a highly conserved sequence, and function may be very similar between members (Das *et al.*, 2015; Pils *et al.*, 2005). In contrast, some protein families may have diverged significantly in sequence and function. Identifying unifying information about the family may be difficult in this case.

3.1.3. Text mining

Text mining is the application of computer algorithms to text in order to extract structured data. The study of algorithms which can be applied to text is within the natural language processing (NLP) field of computer science. I will now summarise some relevant concepts in NLP.

Tokenisation

Tokenisation is the transformation of text into a stream of 'tokens' (Jurafsky and J. H. Martin, 2009). Tokenisation is a way of adding initial structure to unstructured text. Tokenisation is often the first step of a text mining pipeline. Text is usually split on word boundaries, and elements of punctuation may be segmented into tokens too. For example the following sentence

> Tokenisation isn't always straight-forward.

might reasonably be segmented as

> [Tokenisation] [is] [n't] [always] [straight-forward] [.]

A naive algorithm which splits on whitespace will not correctly tokenise punctuation like fullstops and commas, and splitting on punctuation will incorrectly separate hyphenated words. Scientific writing may include mathematical notation such as decimal points, which are identical to fullstops, but where a fullstop indicates a sentence boundary, a number with a decimal point should be output as a single token. In English, the characters for apostrophe/quotation marks are interchangeable, so if the tokenisation above is desired, rules for identifying possessives and contractions need to be included.

The tokenisation algorithm used will depend on the application, and the specific details of the text processing pipeline. For example, some systems for identifying mentions of biological entities in text, such as gene names, may require that one entity maps to one token. In that case, the tokenisation algorithm must not split up gene names, even if they contain whitespace. On the other hand, the system may be able to identify gene names split into multiple tokens, so the tokenisation may be performed by a less sophisticated algorithm (Krallinger *et al.*, 2008).

Stemming

Stemming is the processing of a text to replace all words with their stem word (Uyar, 2009). In most natural languages, words are inflected according to grammatical rules, such as pluralisation and gender. English has fewer inflected forms than many other languages, but there are

also many irregularly inflected words. English nouns are only inflected to plural and possessive forms. For example, the regular possessive noun 'language's' would be stemmed to 'language', and the irregular plural noun 'geese' would be stemmed to 'goose'.

Stemming may be familiar to users of web search engines. The Google search engine will automatically include the stem and other inflections of search terms (Uyar, 2009). Analogously, in biocuration, stemming is a basic tool in literature search to ensure that relevant publications aren't missed. For example, suppose we're interested in finding literature discussing the interaction between the proteins α-syn and VAMP2. A search for 'α-syn VAMP2 interaction' will miss the following two relevant sentences.

> Interestingly, **α-SYN** exists as an unfolded and monomeric cytosolic form and a multimeric membrane-bound form that chaperones the SNARE complex assembly by **interacting** with **VAMP2** [21–23]. *Baksi et al. (2016)*

> **A-syn interacts** with v-SNARE synaptobrevin 2 or **VAMP2** [31], which is a vesicle protein that is involved in mediating fusion of vesicle to the plasma membrane [32,33]. *Lou et al. (2017)*

Indeed, since both sentences use different inflections of 'interact', we would need to use a boolean OR term in order to find both of these sentences with the same search, if our search engine does not perform stemming. Having stemmed these sentences using the Snowball stemmer developed by Porter (2001) we obtain the following[1]:

> interest, **α-syn** exist as an unfold and monomer cytosol form and a multimer membrane-bound form that chaperon the snare complex assembl by **interact** with **vamp2** [21–23].

> **α-syn interact** with v-snare synaptobrevin 2 or **vamp2** [31], which is a vesicl protein that is involv in mediat fusion of vesicl to the plasma membran [32,33].

Hence, a search engine which stems the query text and the corpus text will retrieve both of these documents with a search for 'α-syn VAMP2 interaction'. Stemming is a kind of normalisation, and could be applied during text mining tasks across different domains. More specialised

[1] Specifically, the Snowball stemmer implementation in the Natural Language Toolkit version 3.3 by Bird et al, available from `http://nltk.org/`.

Protein	Synonyms
alpha-synuclein	alpha synuclein alpha-syn α-synuclein α synuclein α-syn
VAMP2	VAMP-2 synaptobrevin-2 synaptobrevin 2

Table 3.1. Most protein names can be rendered in several different ways. Aside from differences in punctuation, a single protein may have two or more completely different names.

normalisation algorithms may be desirable for tasks with biomedical literature. Gene and protein names may be rendered in a variety of different ways (see table 3.1).

A normalisation tool, Norm, designed for use with biomedical entities including protein and gene names is distributed by the NLM as part of their software package, the NLM Lexical Tools (Lu *et al.*, 2005). This tool performs stemming, substitutes spelling variants (e.g., 'haem'/'heme'), removes punctuation and possessives, and normalises the word order.

A related process to stemming is *lemmatisation*. Like stemming, lemmatisation is intended to remove inflections, but in a way which produces the root word as one might find in a dictionary. For example, in the sentence above, the Snowball algorithm stemmed the word 'assembly' to 'assembl'. Lemmatisation would produce the root word 'assemble'. Thus, stemming is a crude form of lemmatisation (Jurafsky and J. H. Martin, 2009).

Named entity recognition

Named entity recognition (NER) is the identification and classification of noun phrases which correspond to 'things' within a category or categories of interest. For example, a common NER task is to identify noun phrases within text which correspond to individuals or locations. In the following sentence, named entities within these categories have been highlighted:

> [Margaret Oakley Dayhoff] was born in [Philadelphia] in 1925.

One of the most frequent tasks within the biomedical literature is to perform NER of biologically relevant entities, such as genes, proteins, taxa, organs, diseases and chemicals. If we performed NER to identify genes and proteins for one of the previous example sentences, we would hope to identify the following.

> Interestingly, α-syn exists as an unfolded and monomeric cytosolic form and a multimeric membrane-bound form that chaperones the SNARE complex assembly by interacting with VAMP2 [21–23].

As noted previously, gene and protein names can be rendered differently by different authors. Gene names may not follow the usual rules of English writing. For example, *p53* will not be capitalised at the beginning of a sentence. Some genes may be given ambiguous names. In the *Drosophila* community, genes are named after mutant phenotypes. For example, the gene *eyeless* or *ey* for short, affects eye development. Hence, *Drosophila* genes are often common English words, rendered lowercase, frustrating the entity recognition process (Hales *et al.*, 2015). Hence, NER within the biomedical literature is a specialised area.

In order to test the effectiveness of different systems, Tanabe *et al.* (2005) published a gold standard corpus of tagged gene/protein names in twenty thousand sentences extracted from PubMed abstracts. Subsequently, researchers have competed to maximise performance over versions of this 'GENETAG' corpus.

Performance measurement

In information retrieval, *precision* and *recall* are the most widely used measures of performance. Precision measures what proportion of the results which are retrieved are correct, and recall measures what proportion of correct results are retrieved.

In the context of NER of genes and proteins, precision measures what proportion of the genes and proteins in the text are identified as such, and recall measures how often the system is correct in its identification of a particular fragment as a gene or protein. Their formal definitions are as follows.

$$\text{precision} = \frac{TP}{TP + FP} \tag{3.1}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{3.2}$$

Where TP is the number of *true positives*, FP is the number of *false positives* and FN is the number of *false negatives*. For a summary of these classifications, see table 3.2. Notice that it is trivial to maximise either one of these metrics. A degenerate system which classifies every token as a gene/protein name will achieve 100% recall, whereas a system which classifies no tokens as a gene/protein name will achieve 100% precision.

Predicted label	True label	
	Positive	*Negative*
Positive	True positive	False positive
Negative	False negative	True negative

Table 3.2. The possible classifications of labelled data for the evaluation of performance.

These two measures are frequently combined into a single score, by taking the harmonic mean of the two. This is called the F_1 score.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.3}$$

In order to evaluate a gene and protein NER system, researchers will often measure its precision, recall and F_1 score against a published gold standard corpus. However, it's important not to neglect the interpretation of performance within the context in which the NER system will be used. For example, in a NER system used in a literature triage system, it may be required that the system recalls a very high proportion of publications discussing a particular protein, so a lower precision may be tolerated. Against the GENETAG corpus, precision and recall of 70-90% are typical for competitive NER systems.

In information retrieval, the usual assumption is that all data falls into two categories: Relevant or irrelevant. In the context of classification, there may be more than two categories. For example, given a publication, perhaps we would like to classify its subject as biology, physics or other. A typical approach to this problem would be to train two classifiers, one for biology publications and one for physics publications. To measure the combined performance of the two classifiers performing this task, we can assess the precision and recall in two different ways. The first method is to sum the *TP*, *TP* and *FN* rate for both classifiers, and calculate precision and recall from these totals. This is the *micro averaged* precision and recall. Alternatively, we can calculate the precision and recall independently for each classifier, and then find the average of the pair of precision and recall values. This is the *macro averaged* precision and recall. The micro and macro averaged precision and recall will be the same if the two classes are equal in size, that is, if there are an equal number of physics and biology publications in the collection. If the categories are unequal in size, the micro averaged precision and recall will be weighted towards the larger category.

Entity linking

Entity linking or entity disambiguation is the grounding of named entities, extracted from text through named entity recognition, to a specific entry in a knowledge base. For the example above, we might wish to link 'Margaret Oakley Dayhoff' and 'Philadelphia' to their respective Wikipedia pages. For biomedical literature, it is desirable to identify entities which can be linked to entries in biological databases. For example, linking mentions of proteins to their UniProt entry.

A complication is posed by metonymy: Gene names are often used by biologists in place of the name of the protein product, and protein complexes may be referred to by a single one of their component proteins (Hunter and Cohen, 2006; C. Li *et al.*, 2014). For example, pyruvate dehydrogenase is a multimeric enzyme which forms part of the pyruvate dehydrogenase complex. This protein may be abbreviated as either PDH, and in the context of the complex as E1. These names can be used ambiguously, as in the following example.

> Pyruvate is the end product of glycolysis and is converted to acetyl-CoA through the activity of the **pyruvate dehydrogenase complex** of enzymes. The acetyl-CoA then directly enters the TCS cycle at citrate synthase where it is combined with oxaloacetate to generate citrate. In metazoans, the conversion of pyruvate to acetyl-CoA is irreversible and therefore represents a critical regulatory point in cellular energy metabolism. **Pyruvate dehydrogenase** is regulated by three known mechanisms: it is inhibited by acetyl-CoA and NADH, it is stimulated by reduced energy in the cell, and it is inhibited by regulatory phosphorylation of its **E1** subunit by pyruvate dehydrogenase kinase (PDK)
>
> *Papandreou et al. (2006)*

The first bolded phrase clearly refers to the complex. We may expect that the second bolded text refers to the PDH subunit of the complex. However, the later reference to 'its E1 subunit' makes clear that the second bolded phrase actually refers to the whole complex.

By convention, gene names are italicised, and their protein product set in upright letters. However, this convention is patchily adhered to, and in any case, text formatting information is often absent from sources of literature, particularly in the case of abstracts. Biologists within different communities may use different systems of nomenclature. For example the aforementioned *Drosophila* gene *eyeless* is known by other biologists as *PAX6*. This naming convention can cause further ambiguities. For example, the gene *hedgehog* may be confused with a species name (Hales *et al.*, 2015). And once again, the gene name may be used to refer to its protein

product. Biologists may change and reuse terminology. Older terminology may be replaced (Bossy *et al.*, 2012).

3.2. Method

3.2.1. Full text search

In order to identify literature which can be curated into a protein family's entry in Pfam, it's necessary to identify mentions of the members of the family within the literature. In Pfam, proteins are identified by their UniProt accession. Hence, entities identified in the literature should be linked to entries in UniProt. Authors rarely cite UniProt accessions, so proteins must be identified based on references to the more ambiguous protein and gene names. As a source of literature, I used the open access subset of PMC. I focussed my initial attention on full-text literature, due to its potential for allowing richer analysis of article content, and its higher accessibility for end-users.

Training data

In order to link proteins to publications, I needed labelled data. The Swiss-Prot subset of UniProt is manually curated, and the publications curated to each entry are listed, and annotated with a 'scope'[2]. The scope is a description of the information which the curator extracted from a particular publication. While this information is written as free text, curators make use of a list of standard topics (UniProt Consortium, 2018). Hence, the publication list for protein entries in Swiss-Prot provides a mapping from literature to protein, and the scope provides a topic for this mapping.

Once a link between proteins and publications has been found, this can form the basis for a list of relevant literature for a HMMER search. However, a HMMER search may return hundreds or thousands of proteins, and these proteins may be mentioned in many different publications. Hence, some method for identifying the literature most useful to a HMMER search user is required. As discussed previously, certain types of papers are more likely than others to be useful to users and curators. Classifying the topic of the paper can therefore form the basis for identifying useful papers. The Swiss-Prot scope can be used to train such a classifier.

I identified all literature within the PubMed Central open access subset which had been cited in a Swiss-Prot entry. Using the scope information, literature was labelled into one of ten categories: *Expression*, *Family & Domains*, *Function*, *Interaction*, *Names*, *Pathology & Biotech*, *PT-*

[2]Equivalent to the RP line in the flat text formatted UniProt entry

M/processing, *Sequences*, *Structure*, and *Subcellular location*. The category is determined by application of regular expressions to the scope annotation. Any scope not falling into these categories is labelled *Unclassified*. Note that a single reference can be assigned multiple scopes. Therefore, an article can be assigned multiple labels. For a full list of scopes and their corresponding label see section A.1. These were supplied by Cecilia Arighi (personal communication, 2017).

Literature processing

I developed a Python package for processing literature and protein database files, which I called 'Grubbler'[3]. The literature is preprocessed by Grubbler before extracting features. The article XML is parsed in order to extract article body text as plain text. Sentence boundary detection and tokenisation is performed using the Stanford CoreNLP toolkit (Manning *et al.*, 2014). Grubbler is available on a public source code repository[4].

Matching literature to proteins

The output of a HMMER search is a set of protein regions, specified by a UniProt identifier and sequence coordinates, and an E-value. In order to identify literature relevant to a particular HMMER search, it is necessary to link entries in the UniProt knowledgebase to mentions of the proteins within the literature. Identifying mentions of proteins in literature is a NER task. I use the BANNER named entity recognition system to identify possible mentions of proteins and genes within the literature (Leaman and Gonzalez, 2008). These extracted entities are then normalised using the NLM Lexical Tools[5]. A cartoon of the literature processing system is shown in figure 3.2.

The UniProt knowledgebase records multiple synonyms for each protein entry. These usually include, at a minimum, a protein name and a gene name. The synonyms for each protein in UniProt are also normalised, and then matched to the extracted entities. This results in a list of candidate mentions for each protein. From a corpus of 1.31 million PMC publications, BANNER extracts 80.4 million entities. After normalisation, the matching of these names with UniProt results in 312 million UniProt-PMC pairs. Some examples of the links made are show in table 3.5. Analysing the entities identified, I noticed that many of the spurious or ambiguous entities identified were three letter alphabetic entities (e.g. 'cas', but not 'p53'). In particular, the word 'fig' was often misidentified as a reference to a protein or gene, indeed, 'fig' was the second most identified entity in PMC, after insulin (see table 3.3). I determined that most of these were

[3] *Grubbler* is an archaic spelling of the word *grubber*, which the Oxford English Dictionary defines as 'One who grubs, lit. and fig.; a digger; a searcher among ruins and the like; a laborious worker'. I felt this an appropriate name for a tool intended to reduce the laborious work of literature search.

[4] `https://bitbucket.org/mjeffryes/grubbler`

[5] https://lsg3.nlm.nih.gov/LexSysGroup/Projects/lvg/current/web/index.html

Entity	Count
insulin	690,423
fig	619,953
cd4	413,214
actin	344,060
gfp	312,803
p53	278,157
nf-κb	230,304
cd8	227,393
ifn	222,804
akt	211,233

Table 3.3. The ten most identified entities in PMC by BANNER.

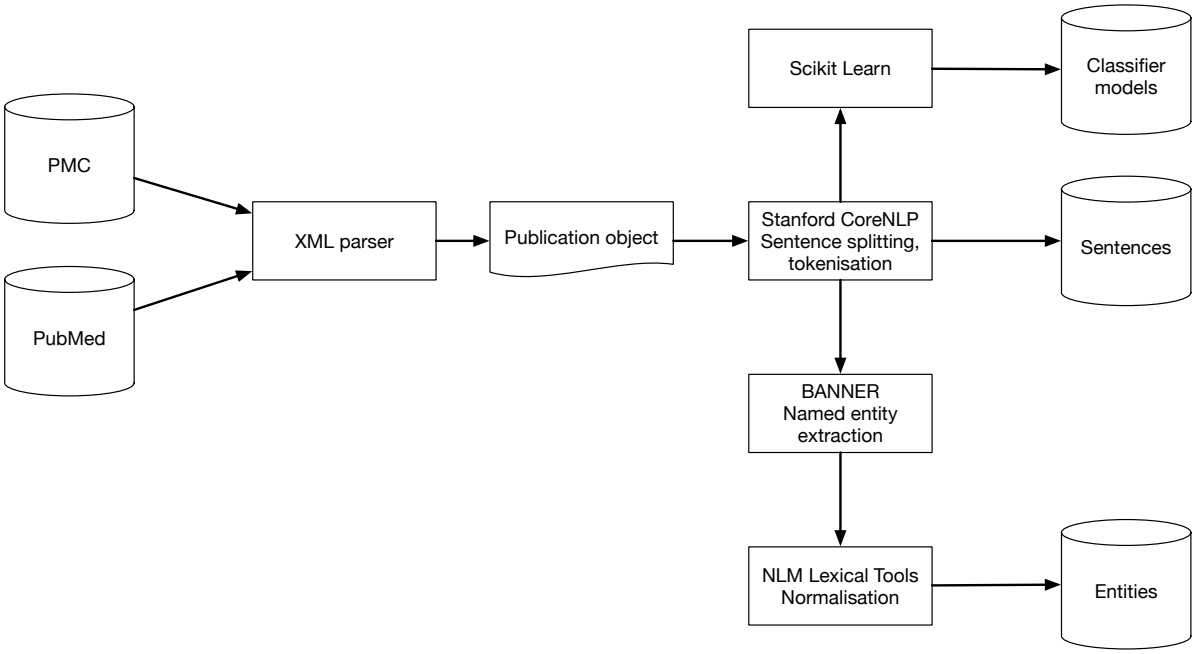


Figure 3.2. A cartoon of the Grubbler literature processing system.

abbreviations of the word 'figure', and not a reference to a protein or gene. Therefore, I filtered the list to remove these three letter alphabetic entities.

In addition, some proteins are mentioned extremely frequently in the literature (see table 3.3). Resources such as FamPlex, a semi-automated resource for linking protein families and complexes to Pfam in Interpro, could assist in linking some entities to their Uniprot entries (Bachman *et al.*, 2018). However, this resource is intended to assist in linking frequently mentioned proteins. Linking these entities to proteins is not especially useful for our application. A protein sequence similarity user who finds that their query sequence is homologous with p53 will not have difficulty finding literature relevant to this protein. Therefore, I limited the matches to literature to 100 for each UniProt entry. In cases where there is a match to more than 100 publications, the 100 which mention the protein most are retained. This reduces the number of pairs to 23 million. I also noticed that orthologues in UniProt often have very similar entries. In cases where two proteins are referred to by identical names, there is no need to treat them separately when matching these names to the literature. Therefore, I 'clustered' UniProt entries where the protein and gene names are identical. This further reduces the number of pairs to 4.9 million.

To measure the performance of the the links identified between UniProt entries and publications in PMC, we can use the references curators have annotated to Swiss-Prot entries. I created a list of ground truth UniProt-PMC pairs by extracting the references from SwissProt. I excluded references which were labelled as `LARGE SCALE` in their scope line, since such publications usually identify many tens of genes or proteins, and may not mention all of them in their text, but perhaps only in their supplementary material. For the same reason, I excluded any pair where the PMC publication was referenced by 70 or more different Swiss-Prot entries. This resulted in 18,083 ground truth pairs. The performance with and without the two filtering methods described is shown in table 3.6. Note that while recall is a reasonable performance metric, precision is not, since Swiss-Prot does not reference all publications which mention a particular protein. We would expect many more references to a protein to be found in the literature than there are publications in a Swiss-Prot entry's publication list. Excluding publications which mention frequently mentioned proteins from consideration has a large effect on recall, but as noted, we are not interested in literature search for these frequently mentioned proteins.

UniProt release `2016_05` was used[6]. PubMed Central does not have versioning, but the maximum PubMed Central identifier (PMCID) included in this experiment was `PMC4898948`.

Once the features are extracted, a binary linear classifier is trained for each label, to discern whether an article is in category or out of category. They are L2-regularised logistic regression

[6] `ftp://ftp.uniprot.org/pub/databases/uniprot/previous_major_releases/release-2016_05/`

Protein	Sentence	Comment
Otoferlin (OTOF_MOUSE)	Mutations in OTOF gene, encoding **otoferlin**, cause DFNB9 deafness and non-syndromic auditory neuropathy (AN). The aim of this study is to identify OTOF mutations in Chinese patients with non-syndromic auditory neuropathy. (Wang *et al.*, 2010)	Correct link to an orthologue.
ATP-PFK (PFKA_CALFI)	**Phosphofructokinase** (Pfk) and Triose phosphate isomerase (Tpi) have FDRs in the RIP-Chip data of 5.15% and 6.08%, respectively, and both are targets of Smaug-mediated transcript degradation and translational repression. (Chen *et al.*, 2014)	Correct link to an orthologue.
Tyrosine aminotransferase (ATTY_DICDI)	The viral **Tat** protein recruits human Super Elongation Complex (SEC) to paused Pol II to overcome this limitation. (Z. Li *et al.*, 2013)	This link is incorrect, the linked protein is Tyrosine aminotransferase, the protein referred to in the sentence is HIV Tat.
Bcar1 (BCAR1_MOUSE)	During this period, rats were fed a standard diet with 13% of casein (**CAS**). (V. Martin *et al.*, 2013)	Incorrect link, proteins have the same abbreviation (cas).
Lipase (LIP_STAEQ)	There is controversy on whether or not hyperinsulinemia directly suppresses BNP production [13,14]. NPs, through the activation of the biologically active membrane guanylate-cyclase-linked NPR-1 has a potent lipolytic effect in human adipocytes via a cGMP-dependent mechanism [15] and activation of hormone-sensitive **lipases** [16,17]. (Nakatsuji *et al.*, 2012)	Lipase is too generic a term to be usefully linked to a single UniProt entry without additional context.
Cellulase 1 (GUN1_STRRE)	Although many hyperthermophilic **endoglucanases** have been reported from archaea and bacteria, a complete survey and classification of all sequences in these species from disparate evolutionary groups, and the relationship between their molecular structures and functions are lacking. (Shi *et al.*, 2014)	Endoglucanases are a class of proteins, the linked protein is not necessarily relevant in this case.
Peptidyl-tRNA hydrolase (PTH_METJA)	Using in vivo microcomputed tomography (micro-CT), we found in parathyroid hormone (**PTH**)-treated osteopenic rats linear increases in cortical and trabecular, due to increased trabecular thickness and number, bone mass. (Brouwers *et al.*, 2009)	This link is incorrect.
Insulin (INS_CARSE)	In contrast to the previously reported $_1$ integrin activation in response to hypoosmolarity or **insulin**, TUDC-induced integrin activation occurs inside the hepatocyte and requires TUDC uptake via the basolateral Na+/taurocholate cotransporting polypeptide (Ntcp) [2]. (Sommerfeld *et al.*, 2014)	Insulin is one of the most frequently mentioned proteins in the literature. Each insulin orthologue could be linked to millions of mentions.

Table 3.5. Eight randomly chosen links between entities identified by BANNER and Swiss-Prot entries, without applying any kind of filtering to the entities.

Filtering	Candidates	TP	Recall
None	311,971,621	13,625	75.6%
Three letter alphabetic	168,829,599	13,029	72.3%
Frequently mentioned	26,402,353	12,147	67.2%
Both	23,411,328	11,842	65.5%

Table 3.6. Recall performance of Swiss-Prot references from PMC using BANNER and normalisation.

classifiers, trained with the SAGA algorithm. Regularisation is a method for reducing overfitting (Ng, 2004). L2 regularisation adds a term to the objective function to minimise the sum of the square of weights. The SAGA algorithm is a variant of the stochastic average gradient algorithm, itself a variant of the classic stochastic gradient descent algorithm (Defazio *et al.*, 2014). In gradient descent, an objective function is minimised by varying the function arguments in the direction which reduces the value of the objective function the fastest, and this is continued until convergence (Schmidt *et al.*, 2013). Stochastic gradient descent is applicable when the objective function is the result of summing objective functions for multiple elements in a data set, such as the error rate of a classifier across many different training samples. In stochastic gradient descent, the function arguments for every training sample are adjusted in the direction that minimises the objective function the most for one randomly chosen sample, rather than calculating this for the entire data set. The stochastic average gradient method further modifies this by moving the weights in a direction which is an average of the direction which was optimal for each training sample, updating the average for one randomly chosen sample at each step (Schmidt *et al.*, 2013). The SAGA method slightly modifies the way the average is updated (Defazio *et al.*, 2014). The classifier was trained using the Scikit-learn Python package (Pedregosa *et al.*, 2012). The SAGA trained linear classifier was chosen because it appears to give the best performance across the categories of most interest.

The previously mentioned MeSH database provides possible features for training classifiers. Entries in PMC are identified by a PMCID, which was mapped to a PubMed identifier (PMID) using the ELink web service provided by the National Centre for Biotechnology Information (NCBI). This enables linking of PMC entries to the bibliographic record in the PubMed database. From PubMed, the MeSH terms for each article can be identified.

The classifiers are trained over several different features, which are calculated from the publication entry in PubMed Central, and its corresponding PubMed citation.

1. The term frequency–inverse document frequency (tf–idf) for each token in the corpus' lexicon, calculated by the *tf* of the word in the publication.

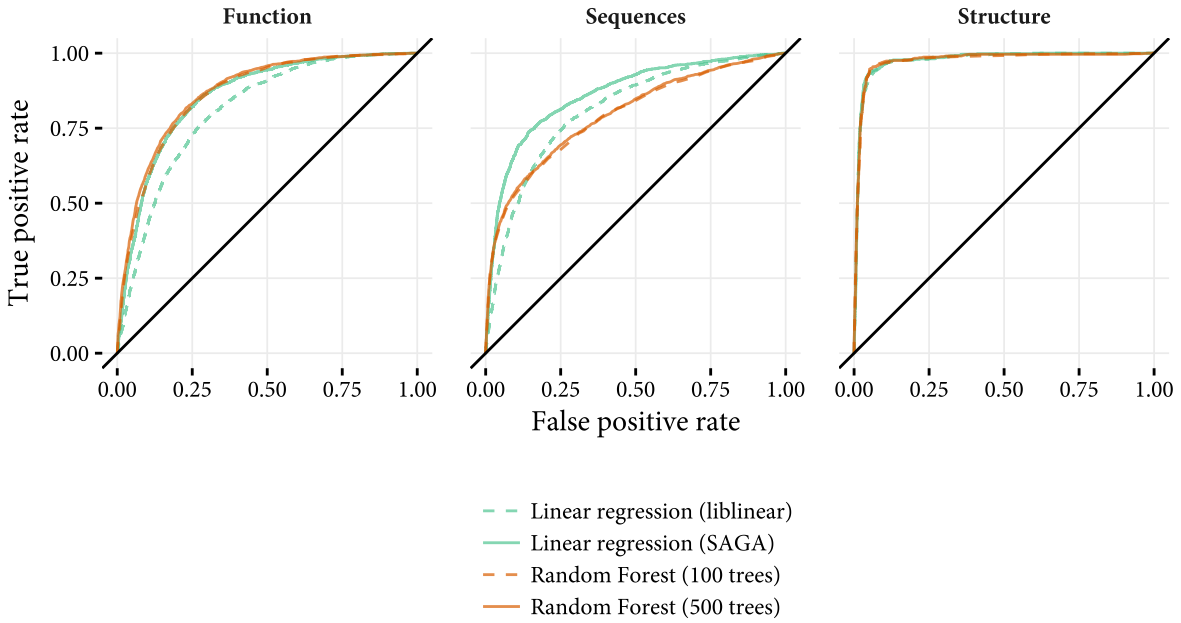


Figure 3.3. Receiver operating characteristic curves for linear classifiers trained with two algorithms and random forest classifiers with 100 and 500 trees, across the three categories of most interest. The data set used is described in section 3.2.2.

Tf–idf is a measure of a particular term's frequency in a document, discounted by its frequency in the corpus. The tf–idf value for a term t in a document d, from a corpus of documents C is calculated as follows:

$$\mathrm{tf}(t,d) = \begin{cases} 1+\log(t_d), & \text{if } t_d > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

$$\mathrm{idf}(t,C) = \log\left(\frac{1+|C|}{\sum_{d\in C} t_d}\right) \tag{3.5}$$

$$\text{tf-idf}(t,d,C) = \mathrm{tf}(t,d)\cdot\mathrm{idf}(t,C) \tag{3.6}$$

Where t_d is the number of instances of t found in d.

This is represented as a vector d, the size of the corpus' lexicon, with the d_i being the tf–idf of the ith word in the corpus' lexicon. The resulting vector has 341,622 elements.

2. The MeSH terms of the article. MeSH terms are divided into descriptors and qualifiers. Descriptors are placed within a tree taxonomy. For example, the descriptor 'Genetics' is placed within 'Biology'. Each descriptor may be modified by one or more qualifier terms. Each descriptor has a list of permitted qualifiers. For example, the qualifiers for genetics include 'ethics' and 'standards'. Descriptors are sometimes called headings, and qualifiers are sometimes called subheadings. Each term, descriptor or qualifier, can be flagged as 'major' which denotes that it is a main topic of the article. Note that a qualifier can be a major term, even if its descriptor is not, and vice versa.

   The MeSH terms are represented by two vectors. One vector with each element set to 1 if the corresponding MeSH term is assigned to the publication, otherwise 0, and the other vector with each element set to 1 if the corresponding MeSH term is assigned as a major term to the publication, otherwise 0. The former vector has 12,441 elements, the latter 2526.

3. The paragraph count of the publication. One integer.

These three features are concatenated into one feature vector, for each publication.

The open access subset of PubMed Central contains 1.2 million articles. Of these, 90% were allocated to the training set, and 5% each to the development and testing sets.

3.2.2. Abstract search

I identified that limiting search to only full-text PubMed Central publications could eliminate a large proportion of potentially relevant literature. While it might be desirable to return open access literature to the user, in some cases, none will be available. The vast majority of the biomedical literature is deposited in MEDLINE, which is accessible through the PubMed web service. An abstract is available for almost all current literature deposited in PubMed. Therefore, I decided to train a set of classifiers, similar to those described above, to categorise abstracts. These classifiers were trained over only tf features (see equation (3.4)), since MeSH terms are not available for all publications in PubMed.

PubMed contains 27 million citation records. Many of these, particularly the older records are missing important data, such as abstract text. For classifier training, I sampled only from English language journal articles, reviews and letters published since the year 2000, where the abstract was present. Of these, approximately 120,000 entries had been cited by an entry in Swiss-Prot. I sampled 120,000 further entries from the remainder, to give approximately 240,000 entries. These additional entries were sampled to ensure that the classifier was not

trained only on literature referenced by Swiss-Prot, which could lead to poor performance on the wider biomedical literature. Once again, 90% were allocated to the training set, and 5% each to the development and testing sets.

3.3. Results

3.3.1. Category classifiers

Shown in tables 3.4b and 3.5b are the retrieval performance of the PMC and PubMed classifiers respectively. The two results are not directly comparable, since the two experiments used a different testing set composition. The full-text PMC experiment has many fewer positively labeled articles. Indeed, this sparsity of data partially motivated the second experiment on PubMed. Since the vast majority of articles referenced by Swiss-Prot are present in PubMed, more labelled training and testing data is available. In addition, the number of negatively labeled examples in the PubMed experiment is artificially reduced through the sampling technique described in the methods section.

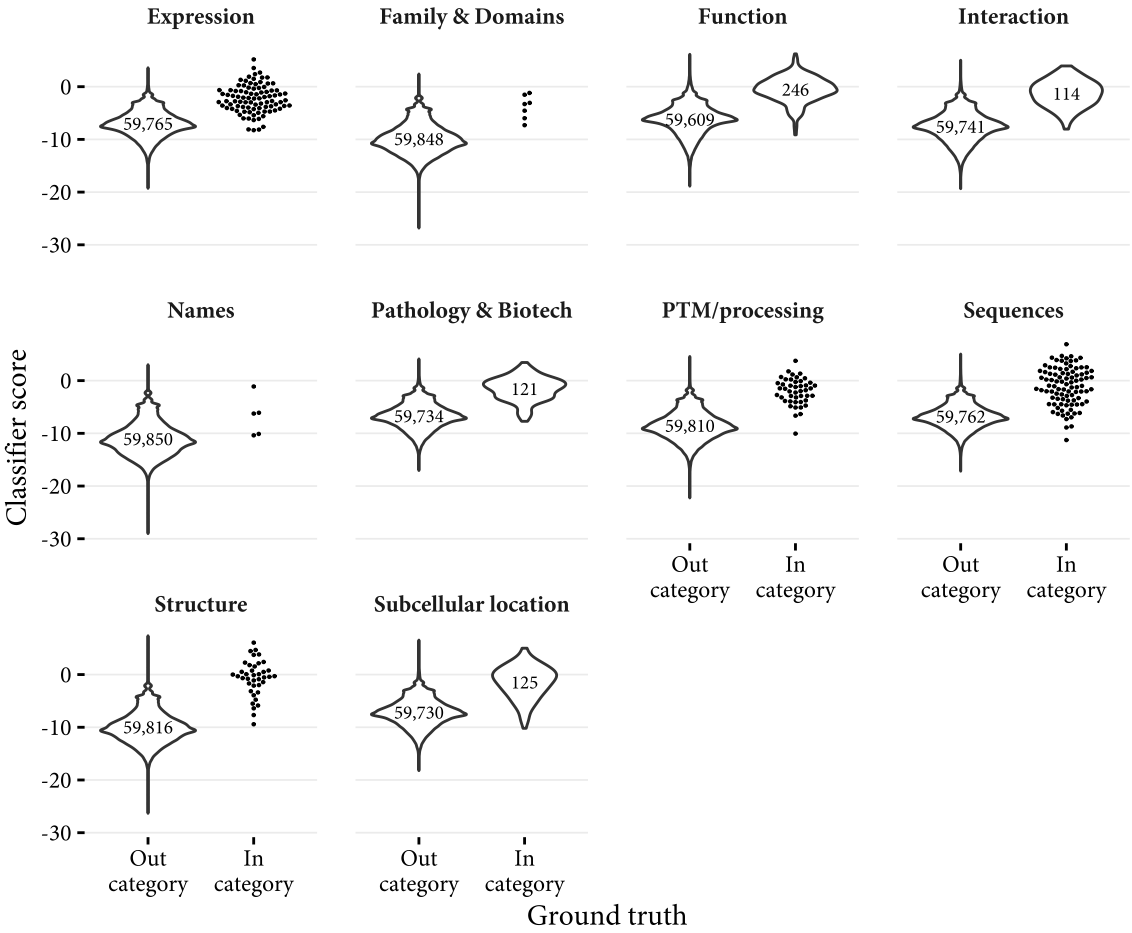
There is significant overlap of classifier scores between the in and out of category publications. However, not all publications which could be referenced by UniProt in a given category are referenced by UniProt. That is, UniProt's references are not a complete categorisation of the entire literature. The purpose of the classifiers is to identify literature, yet uncited by UniProt, which could be annotated to Pfam families.

3.4. Conclusion

The classifiers were developed in order to provide a potential method to rank literature which has been matched to UniProt entries by a basic entity linking system. This in turn can be used to identify curatable literature for new protein families.

The first set of classifiers developed using the data from PMC did not perform well in identifying literature of the relevant category. They produced many false positives, and false negatives. With precision around 0.1, relevant literature would be swamped by irrelevant results, making them unsuitable for their intended purpose.

On the other hand, the second set of classifier developed using the data from PubMed had precision and recall over 0.65 for the function and structure classifiers which I believe are most relevant to identifying curatable literature. Since the task is to assist in the curation of literature,

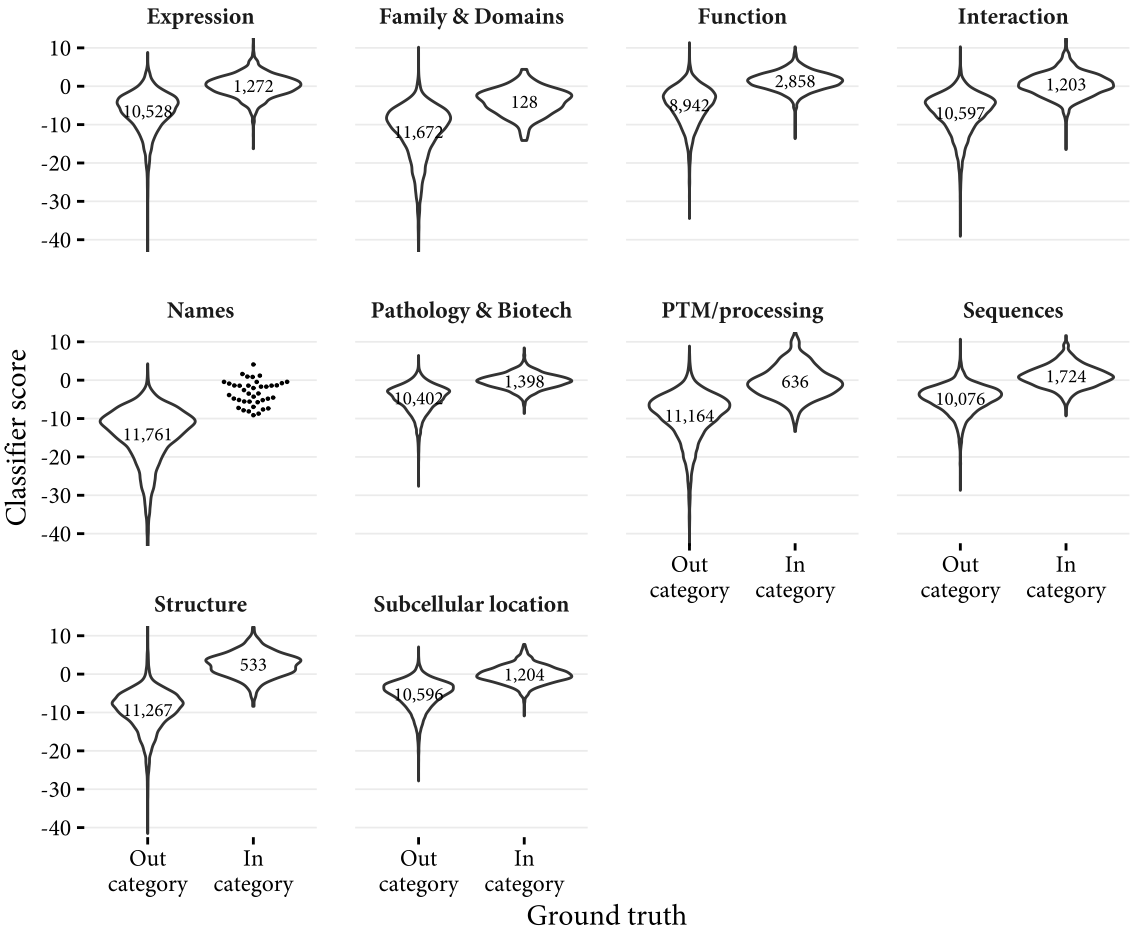


a Distribution of classifier decision scores for the 10 category classifiers trained on the PMC data, for in category and out of category training data. The in category training data are publications which have been referenced by UniProt for the given category. The out of category training data are publications which are not referenced by UniProt or which are referenced by UniProt in a different category. The data are plotted as a beeswarm plot if there are fewer than 100 points, and a density plot otherwise.

Classifier	TP	TN	FP	FN	Precision	Recall	F1
Expression	17	59,610	155	73	0.1	0.19	0.13
Family & Domains	0	59,836	12	7	0	0	0
Function	102	58,902	707	144	0.13	0.41	0.19
Interaction	38	59,541	200	76	0.16	0.33	0.22
Names	0	59,835	15	5	0	0	0
PTM/processing	9	59,720	90	36	0.09	0.2	0.13
Pathology & Biotech	31	59,514	220	90	0.12	0.26	0.17
Sequences	41	59,528	234	52	0.15	0.44	0.22
Structure	16	59,699	117	23	0.12	0.41	0.19
Subcellular location	46	59,485	245	79	0.16	0.37	0.22

b True positive, true negative, false positive, false negative, precision, recall and F_1 scores for the 10 category classifiers trained on the PMC data.

Figure 3.4.



a Distribution of classifier decision scores for the 10 category classifiers trained on the PubMed abstracts data, for in category and out of category training data. The in category training data are publications which have been referenced by UniProt for the given category. The out of category training data are publications which are not referenced by UniProt or which are referenced by UniProt in a different category. The data are plotted as a beeswarm plot if there are fewer than 100 points, and a density plot otherwise.

Classifier	TP	TN	FP	FN	Precision	Recall	F1
Expression	681	9,903	625	591	0.52	0.54	0.53
Family & Domains	12	11,587	85	116	0.12	0.09	0.11
Function	2,106	7,829	1,113	752	0.65	0.74	0.69
Interaction	678	10,153	444	525	0.6	0.56	0.58
Names	5	11,742	19	34	0.21	0.13	0.16
PTM/processing	271	10,926	238	365	0.53	0.43	0.47
Pathology & Biotech	619	9,868	534	779	0.54	0.44	0.49
Sequences	1,116	9,516	560	608	0.67	0.65	0.66
Structure	431	11,073	194	102	0.69	0.81	0.74
Subcellular location	565	10,217	379	639	0.6	0.47	0.53

b True positive, true negative, false positive, false negative, precision, recall and F_1 scores for the 10 category classifiers trained on the PubMed data.

Figure 3.5.

rather than to automatically assign literature to a particular entry, it is tolerable that there are some false positives, since these can be manually filtered out by the curator of the new entry.

As previously noted, the poor performance of the PMC classifiers could be due to the sparsity of the data. The training set for the abstract classifier had around twenty times more samples for each of the categories.

Similar work in categorising publications using UniProt scopes for curation of literature into Swiss-Prot has been conducted by Patrick Ruch and colleagues. They used a variety of machine learning techniques. The F1 scores reported were between 0.66 and 0.80, of a similar range to those of the function and structure classifiers developed here (Teodoro *et al.*, 2017).

More useful scores may be possible if the context of protein mentions is taken into account. For example, determining whether protein function is discussed in sentences surrounding the mention of the protein.

Scores could also take into account whether the sentence discussing a protein is going over previously known facts referenced to other publications, or is making new claims. Previous research has attempted to classify sentences as 'introduction', 'method', 'results' or 'discussion' (Agarwal and Yu, 2009; McKnight and Srinivasan, 2003). It's possible that curators may want to find publications where novel claims are being made, so may want to find mentions of proteins in the results section. Conversely, they may wish to find publications discussing protein function in the introductory sentences, since they may summarise established knowledge.

In the next chapters I will apply these methods to develop literature curation tools.

3.5. References

Agarwal, S. and H. Yu. 'Automatically classifying sentences in full-text biomedical articles into Introduction, Methods, Results and Discussion.' In: *Bioinformatics (Oxford, England)* 25.23 (Dec. 2009), pp. 3174–3180. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btp548. PUBMED: 19783830.

Bachman, J. A., B. M. Gyori and P. K. Sorger. 'FamPlex: a resource for entity recognition and relationship resolution of human protein families and complexes in biomedical text mining'. In: *BMC Bioinformatics* 19.1 (Dec. 2018), p. 248. ISSN: 1471-2105. DOI: 10.1186/s12859-018-2211-5.

Bossy, R., J. Jourde, A.-P. Manine, P. Veber, E. Alphonse, M. van de Guchte, P. Bessières and C. Nédellec. 'BioNLP Shared Task - The Bacteria Track'. In: *BMC bioinformatics* 13.Suppl 11 (2012), S3. DOI: 10.1186/1471-2105-13-S11-S3. PUBMED: 22759457.

Coletti, M. H. and H. L. Bleich. 'Medical Subject Headings Used to Search the Biomedical Literature'. In: *Journal of the american medical informatics association* 8.4 (July 2001), pp. 317–323. ISSN: 1067-5027. DOI: 10.1136/jamia.2001.0080317. PUBMED: 11418538.

Das, S., N. L. Dawson and C. A. Orengo. 'Diversity in protein domain superfamilies.' In: *Current opinion in genetics & development* 35 (Dec. 2015), pp. 40–49. ISSN: 1879-0380. DOI: 10.1016/j.gde.2015.09.005. PUBMED: 26451979.

Defazio, A., F. Bach and S. Lacoste-Julien. *SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives*. July 2014. arχiv: 1407.0202.

Europe PMC Consortium. 'Europe PMC: a full-text literature database for the life sciences and platform for innovation.' In: *Nucleic acids research* 43.D1 (Jan. 2015), pp. D1042–D1048. ISSN: 1362-4962. DOI: 10.1093/nar/gku1061. PUBMED: 25378340.

Hales, K. G., C. A. Korey, A. M. Larracuente and D. M. Roberts. 'Genetics on the Fly: A Primer on the Drosophila Model System.' In: *Genetics* 201.3 (Nov. 2015), pp. 815–842. ISSN: 1943-2631. DOI: 10.1534/genetics.115.183392. PUBMED: 26564900.

Hunter, L. and K. B. Cohen. 'Biomedical Language Processing: Perspective What's Beyond PubMed?' In: *Molecular cell* 21.5 (Mar. 2006), pp. 589–594. DOI: 10.1016/j.molcel.2006.02.012. PUBMED: 16507357.

Jurafsky, D. and J. H. Martin. *An Introduction to Natural Language Processing , Computational Linguistics, and Speech Recognition*. Pearson, 2009. ISBN: 9780131873216.

Krallinger, M., A. Valencia and L. Hirschman. 'Linking genes to literature: text mining, information extraction, and retrieval applications for biology.' In: *Genome biology* 9.Suppl 2 (2008), S8. DOI: 10.1186/gb-2008-9-s2-s8. PUBMED: 18834499.

Leaman, R. and G. Gonzalez. 'BANNER: an executable survey of advances in biomedical named entity recognition.' In: *Pacific Symposium on Biocomputing*. Jan. 2008, pp. 652–663. DOI: 10.1142/9789812776136_0062. PUBMED: 18229723.

Li, C., M. Liakata and D. Rebholz-Schuhmann. 'Biological network extraction from scientific literature: state of the art and challenges.' In: *Briefings in bioinformatics* 15.5 (Sept. 2014), pp. 856–877. ISSN: 1477-4054. DOI: 10.1093/bib/bbt006. PUBMED: 23434632.

Lu, C. J., G. Divita and A. C. Browne. *Enhanced Normalization of Parenthetic Plural Forms: (s), (es), (ies)*. Tech. rep. Bethesda, MD: National Library of Medicine, 2005. URL: https://lexsrv3.nlm.nih.gov/Specialist/Docs/Papers/index.html.

Manning, C., M. Surdeanu, J. Bauer, J. Finkel, S. Bethard and D. McClosky. 'The Stanford CoreNLP Natural Language Processing Toolkit'. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: System demonstrations*. 2014, pp. 55–60. ISBN: 9781941643006. DOI: 10.3115/v1/P14-5010.

McKnight, L. and P. Srinivasan. 'Categorization of sentence types in medical abstracts.' In: *AMIA ... Annual Symposium proceedings. AMIA Symposium* 2003 (2003), pp. 440–4444. ISSN: 1942-597X. PUBMED: 14728211.

NCBI Resource Coordinators. 'Database resources of the National Center for Biotechnology Information.' In: *Nucleic acids research* 46.D1 (Jan. 2018), pp. D8–D13. ISSN: 1362-4962. DOI: 10.1093/nar/gkx1095. PUBMED: 29140470.

Ng, A. Y. 'Feature selection, L1 vs. L2 regularization, and rotational invariance'. In: *Twenty-first international conference on Machine learning*. New York, New York, USA: ACM Press, 2004, p. 78. ISBN: 1581138285. DOI: 10.1145/1015330.1015435.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12 (2012), pp. 2825–2830. ISSN: 1532-4435. arχiv: 1201.0490.

Pils, B., R. R. Copley and J. Schultz. 'Variation in structural location and amino acid conservation of functional sites in protein domain families.' In: *BMC bioinformatics* 6 (Aug. 2005), p. 210. ISSN: 1471-2105. DOI: 10.1186/1471-2105-6-210. PUBMED: 16122386.

Porter, M. *Snowball: A language for stemming algorithms*. Oct. 2001. URL: http://snowball.tartarus.org/texts/introduction.html (visited on 29/06/2018).

Poux, S., C. N. Arighi, M. Magrane, A. Bateman, C.-H. Wei, Z. Lu, E. Boutet, H. Bye-A-Jee, M. L. Famiglietti, B. Roechert and The UniProt Consortium. 'On expert curation and scalability: UniProtKB/Swiss-Prot as a case study'. In: *Bioinformatics* 33.21 (Nov. 2017), pp. 3454–3460. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx439. PUBMED: 29036270.

Schmidt, M., N. L. Roux and F. Bach. 'Minimizing Finite Sums with the Stochastic Average Gradient'. In: (Sept. 2013). arχiv: 1309.2388.

Tanabe, L., N. Xie, L. H. Thom, W. Matten and W. J. Wilbur. 'GENETAG: a tagged corpus for gene/protein named entity recognition.' In: *BMC bioinformatics* 6 Suppl 1.Suppl 1 (2005), S3. ISSN: 1471-2105. DOI: 10.1186/1471-2105-6-S1-S3. PUBMED: 15960837.

Teodoro, D., L. Mottin, J. Gobeill, C. Arighi, P. Ruch, D. Teodoro, L. Mottin, J. Gobeill, C. Arighi and P. Ruch. *Assessing text embedding models to assign UniProt classes to scientific literature*. Sept. 2017. DOI: 10.7490/F1000RESEARCH.1114865.1.

UniProt Consortium. *UniProt Help: Publications section*. URL: https://www.uniprot.org/help/publications_section (visited on 18/07/2018).

Uyar, A. 'Google stemming mechanisms'. In: *Journal of Information Science* 35.5 (Oct. 2009), pp. 499–514. ISSN: 0165-5515. DOI: 10.1177/1363459309336801.

3.6. Examples

Baksi, S., A. K. Tripathi and N. Singh. 'Alpha-synuclein modulates retinal iron homeostasis by facilitating the uptake of transferrin-bound iron: Implications for visual manifestations of Parkinson's disease.' In: *Free radical biology & medicine* 97 (2016), pp. 292–306. ISSN: 1873-4596. DOI: 10.1016/j.freeradbiomed.2016.06.025. PUBMED: 27343690.

Brouwers, J. E., B. van Rietbergen, R. Huiskes and K. Ito. 'Effects of PTH treatment on tibial bone of ovariectomized rats assessed by in vivo micro-CT'. In: *Osteoporosis international* 20.11 (Nov. 2009), pp. 1823–1835. DOI: 10.1007/s00198-009-0882-5. PUBMED: 19262974.

Chen, L., J. G. Dumelie, X. Li, M. H. Cheng, Z. Yang, J. D. Laver, N. U. Siddiqui, J. T. Westwood, Q. Morris, H. D. Lipshitz and C. A. Smibert. 'Global regulation of mRNA translation and stability in the early Drosophila embryo by the Smaug RNA-binding protein'. In: *Genome biology* 15.1 (Jan. 2014), R4. DOI: 10.1186/gb-2014-15-1-r4. PUBMED: 24393533.

Li, Z., J. Guo, Y. Wu and Q. Zhou. 'The BET bromodomain inhibitor JQ1 activates HIV latency through antagonizing Brd4 inhibition of Tat-transactivation'. In: *Nucleic acids research* 41.1 (Jan. 2013), pp. 277–287. DOI: 10.1093/nar/gks976. PUBMED: 23087374.

Lou, X., J. Kim, B. J. Hawk and Y.-K. Shin. 'α-Synuclein may cross-bridge v-SNARE and acidic phospholipids to facilitate SNARE-dependent vesicle docking.' In: *The biochemical journal* 474.12 (June 2017), pp. 2039–2049. ISSN: 1470-8728. DOI: 10.1042/BCJ20170200. PUBMED: 28495859.

Martin, V., S. Ratel, J. Siracusa, P. Le Ruyet, I. Savary-Auzeloux, L. Combaret, C. Guillet and D. Dardevet. 'Whey proteins are more efficient than casein in the recovery of muscle functional properties following a casting induced muscle atrophy'. In: *PLoS ONE* 8.9 (2013), e75408. DOI: 10.1371/journal.pone.0075408. PUBMED: 24069411.

Nakatsuji, H., K. Kishida, T. Funahashi, T. Nakagawa and I. Shimomura. 'Hyperinsulinemia correlates with low levels of plasma B-type natriuretic peptide in Japanese men irrespective of fat distribution'. In: *Cardiovascular diabetology* 11 (Mar. 2012), p. 22. DOI: 10.1186/1475-2840-11-22. PUBMED: 22397400.

Papandreou, I., R. A. Cairns, L. Fontana, A. L. Lim and N. C. Denko. 'HIF-1 mediates adaptation to hypoxia by actively downregulating mitochondrial oxygen consumption'. In: *Cell Metabolism* 3.3 (2006), pp. 187–197. ISSN: 1550-4131. DOI: https://doi.org/10.1016/j.cmet.2006.01.012.

Shi, H., Y. Zhang, L. Wang, X. Li, W. Li, F. Wang and X. Li. 'Molecular analysis of hyperthermophilic endoglucanase Cel12B from Thermotoga maritima and the properties of its functional

residues'. In: *BMC structural biology* 14 (Feb. 2014), p. 8. DOI: 10.1186/1472-6807-14-8. PUBMED: 24529187.

Sommerfeld, A., P. G. Mayer, R. Reinehr and D. Häussinger. 'Mechanisms of tauroursodeoxycholate-mediated inhibition of apoptosis'. In: *European journal of medical research* 19.Suppl 1 (2014), S22–S22. ISSN: 0949-2321. DOI: 10.1186/2047-783X-19-S1-S22.

Wang, D. Y., Y. C. Wang, D. Weil, Y. L. Zhao, S. Q. Rao, L. Zong, Y. B. Ji, Q. Liu, J. Q. Li, H. M. Yang, Y. Shen, C. Benedict-Alderfer, Q. Y. Zheng, C. Petit and Q. J. Wang. 'Screening mutations of OTOF gene in Chinese patients with auditory neuropathy, including a familial case of temperature-sensitive auditory neuropathy'. In: *BMC medical genetics* 11 (May 2010), p. 79. DOI: 10.1186/1471-2350-11-79. PUBMED: 20504331.

4. Implementing combined protein family identification and literature search

The work of the second chapter of this thesis concerns methods for quickly identifying high quality novel protein families, and improvements to existing families. The third chapter concerns the identification of literature relevant to protein family curation. In this chapter I will describe the application of these methods to the development of a prototype application for the identification of novel and improved protein families, and for the curation of novel families.

The web application which I developed to explore these methods is called 'Search-Sifter'. It is designed to have a simple user interface which allows users to perform protein sequence similarity searches, display the relationship between the result sets of these searches and the families in Pfam, and to show literature relevant to the proteins in the search.

4.1. Background

The fusion of protein sequence similarity search with literature search has been investigated previously. Jaroszewski *et al.* (2014) surveyed previous systems, and identified MineBlast, GeneReporter, METIS, and quickLit, along with their own PubServer (Bartsch *et al.*, 2011; Dieterich *et al.*, 2005; Gilchrist *et al.*, 2008; Mitchell, Divoli *et al.*, 2005).

MineBlast and its successor, GeneReporter, perform searches using various BLAST-based algorithms, extract protein and gene names from the search results via the UniProt database, and then use these names to search PubMed. GeneReporter also provides additional options to filter the results and to add keywords to try and identify species specific literature matches (Bartsch *et al.*, 2011; Dieterich *et al.*, 2005).

The METIS system performs a BLAST search with a query sequence and then identifies the literature cited by any Swiss-Prot entries which are found by the search, and identifies terms from the UniProt entries found by the search which are then used to search PubMed. It also uses sentence-based classifiers to identify sentences relevant to structure, function and disease,

but the authors found the performance of these classifiers to be disappointing (Mitchell, Divoli *et al.*, 2005). This system was later refined in Minotaur (Mitchell, Selimas *et al.*, 2012).

The PaperBLAST system developed by Price and Arkin (2017) uses BLAST to search for sequences similar to a query sequence, and then performs a Europe PMC search for a subset of the identifiers of proteins found in the search results. However, the authors note that many publications which mention a protein may not be useful for curation and that 'articles about gene expression often include tables of upregulated or downregulated genes. Because these articles mention many genes, they are overrepresented in the PaperBLAST results'.

The system described in this chapter differs in several ways from these other works. Search-Sifter exploits Swiss-Prot annotations for identifying relevant literature, by using the scope annotations to Swiss-Prot references to train category classifiers, as described in the previous chapter. It presents a comparison of searches to the existing families in Pfam, allowing them to be assessed for novelty. And it uses the HMMER web server for protein sequence similarity search, providing fast, sensitive homology searches (Finn *et al.*, 2011).

4.2. Technical implementation

The prototype Search-Sifter system is operational on the European Bioinformatics Institute (EMBL-EBI)'s web production infrastructure. The system is collectively called 'Search Sifter'. A cartoon of the system is shown in figure 4.1. It is composed of the following modules:

`hmmer-web` Interacts with the HMMER web service application programming interface (API) to perform and retrieve HMMER protein sequence similarity searches.

`grubbler` Processes literature Extensible Markup Language (XML) documents to extract features, and UniProt XML documents to label literature. Trains linear classifier using specified features.

`grubbler-service` Identifies literature relevant to a HMMER search by incorporating classifier scores from `grubbler`, weighted by the E-value and number of proteins mentioned in the literature, and exposes this information via a REST interface.

`search-sifter` Calculates overlap between HMMER search results and Pfam families, using MinHash derived Jaccard containment estimates.

`search-sifter-web` Integrates `hmmer-web`, `search-sifter` and `grubbler-service` and provides a browser based user interface. Users are able to initiate HMMER searches, or analyse an existing search. The analysis results are presented alongside literature relevant to the search, and

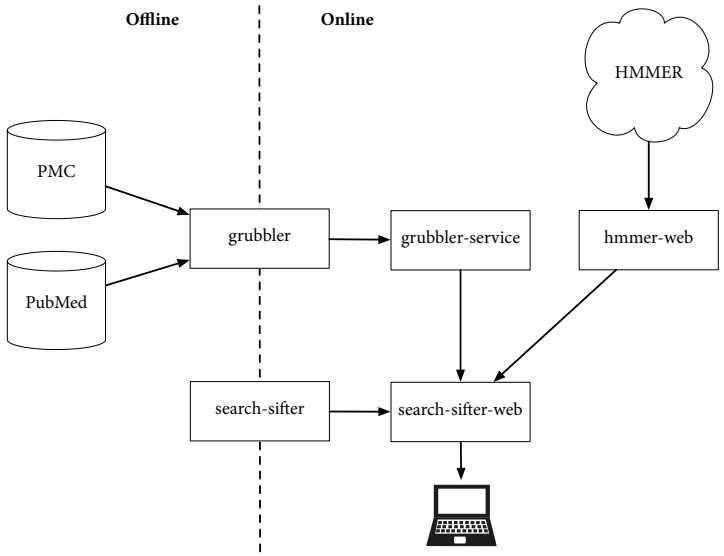


Figure 4.1. A cartoon of the programs involved in the Search-Sifter web application. Literature is processed offline by Grubbler, and family hashes are calculated offline by Search-Sifter. Search-Sifter Web performs HMMER searches with the HMMER API via `hmmer-web`.

families can be submitted to Pfam using a form email containing a link to the Search-Sifter results, the size of the potential new family, and a Stockholm formatted alignment.

4.2.1. Search-Sifter

Search-Sifter is implemented as a Python web application. It uses the HMMER web service API to perform and retrieve searches. Users are able to enter a protein sequence to perform a HMMER sequence similarity search, or enter a HMMER search URL to retrieve an existing search. Users are also able to search using a random protein sequence retrieved from UniProt.

Once the search results are retrieved, they are analysed for overlap with the existing Pfam families, using the technique for estimating Jaccard containment described in chapter 2. The hashes for existing Pfam families are precalculated, so that this step can be performed very quickly. Indeed, the slowest part of the operation is retrieving the search results from HMMER. If there is predicted overlap, the results are presented to the user in the form of a table giving the estimated overlap with existing families, and a size-proportional Euler diagram. An Euler diagram is similar to a Venn diagram, but with only non-zero relationships between sets shown. The size of the circle representing each family is proportional to the size of the family.

In addition to the family analysis, relevant literature is shown on the right side of the page. In the initial iteration of the application, the results were from PubMed Central (PMC). However,

after discussion with potential users, I decided that these results were not complete enough to be useful. Therefore, as discussed in the previous chapter, I trained the classifier on PubMed data too. This provided more comprehensive and useful search results.

The literature search is performed by a second web application, Grubbler, which is described in the next section.

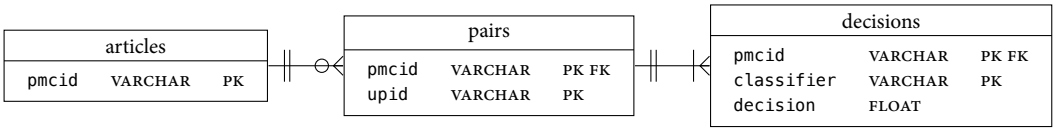


Figure 4.2. An entity relationship diagram showing the schema of the Grubbler SQLite database.

4.2.2. Grubbler

The initial version of the application used the SQLITE library, and a very simple database schema. This enables the application to be deployed without a database server. The only information stored was the unique identifiers of literature in the database, the PubMed Central identifier (PMCID), the proteins identified using BANNER as a PMCID/UniProt name pair, and the classifier score. The entity relationship diagram of this schema is shown in figure 4.2. Instead of serving literature directly, the bibliographic information and abstracts are retrieved asynchronously using the Europe PMC API.

The second iteration of Grubbler, developed to support the Search-Sifter application, and the application discussed in the next chapter, serves literature directly, in order to support more detailed information about literature matches, such as the sentence context around the mention of the protein. This version is deployed with a MySQL database server, with the schema shown in figure 4.3.

The `literature` table in the database stores the bibliographic information about publications. Each row corresponds to a row in either the `pmc_articles` or `medline_abstracts` table. Sentences extracted by the Stanford CoreNLP library are given a universally unique identifier (UUID) and stored in the `pmc_sentences` table. Entities extracted by BANNER are stored in `pmc_entities` and linked to the sentence in which the entity was found. Publications have also been processed to identify any named taxa using the Lineueus library, and these are stored in the `pmc_taxons` table.

The `up_entries` table stores entries from the UniProt database. An entry's identifiers are stored in the `up_identifiers` table. The type of the identifier—that is, whether the identifier is the name of the protein described in the entry, or the gene coding for it—is stored too, and the

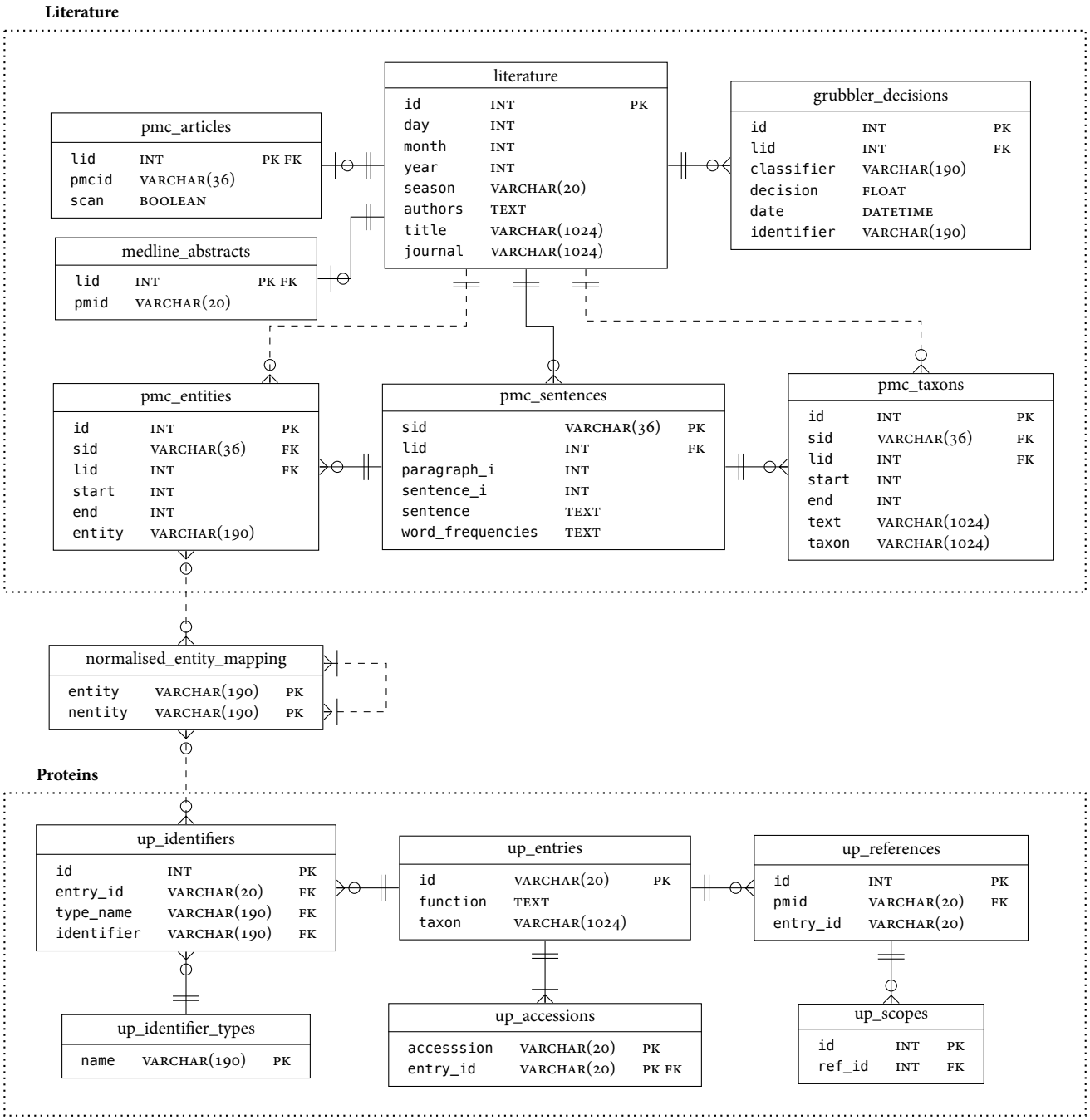


Figure 4.3. An entity relationship diagram of the schema of the Grubbler MySQL database.

possible types of identifier are stored in the `up_identifier_types` tables. The entry's one or more accessions are stored in the `up_accessions` table. The publications cited by the curator of the entry, if applicable, are stored in the `up_references` table. The one or more scopes which the curator annotated to the reference are stored in the `up_scopes` table.

These two halves of the database are linked together by the `normalised_entity_mapping` table. Every identifier in UniProt, and entity found in the literature are normalised using the National Library of Medicine (NLM) lexical tools, and the resulting pairs of names, before and after normalisation, are stored in the table.

The literature search sorts literature which mentions proteins in the search results by the score of the function classifier, described in the previous chapter.

The *Function* classifier is in use to rank literature as a component of the `search-sifter-web` module of the prototype system. The user interface of the Search-Sifter web application is shown in figures 4.4 and 4.5.

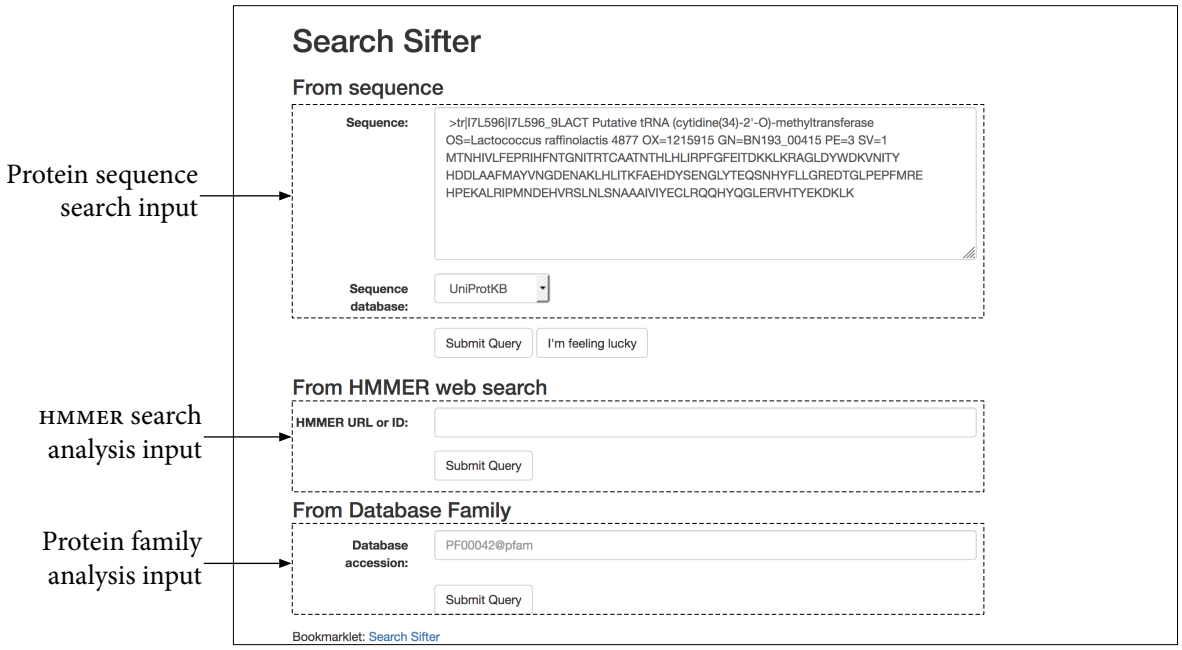


Figure 4.4. Screen capture of the Search-Sifter input user interface. Users can analyse a new search by inputing a protein sequence, an existing HMMER web search by inputting its URL, or an existing protein family by inputting its accession.

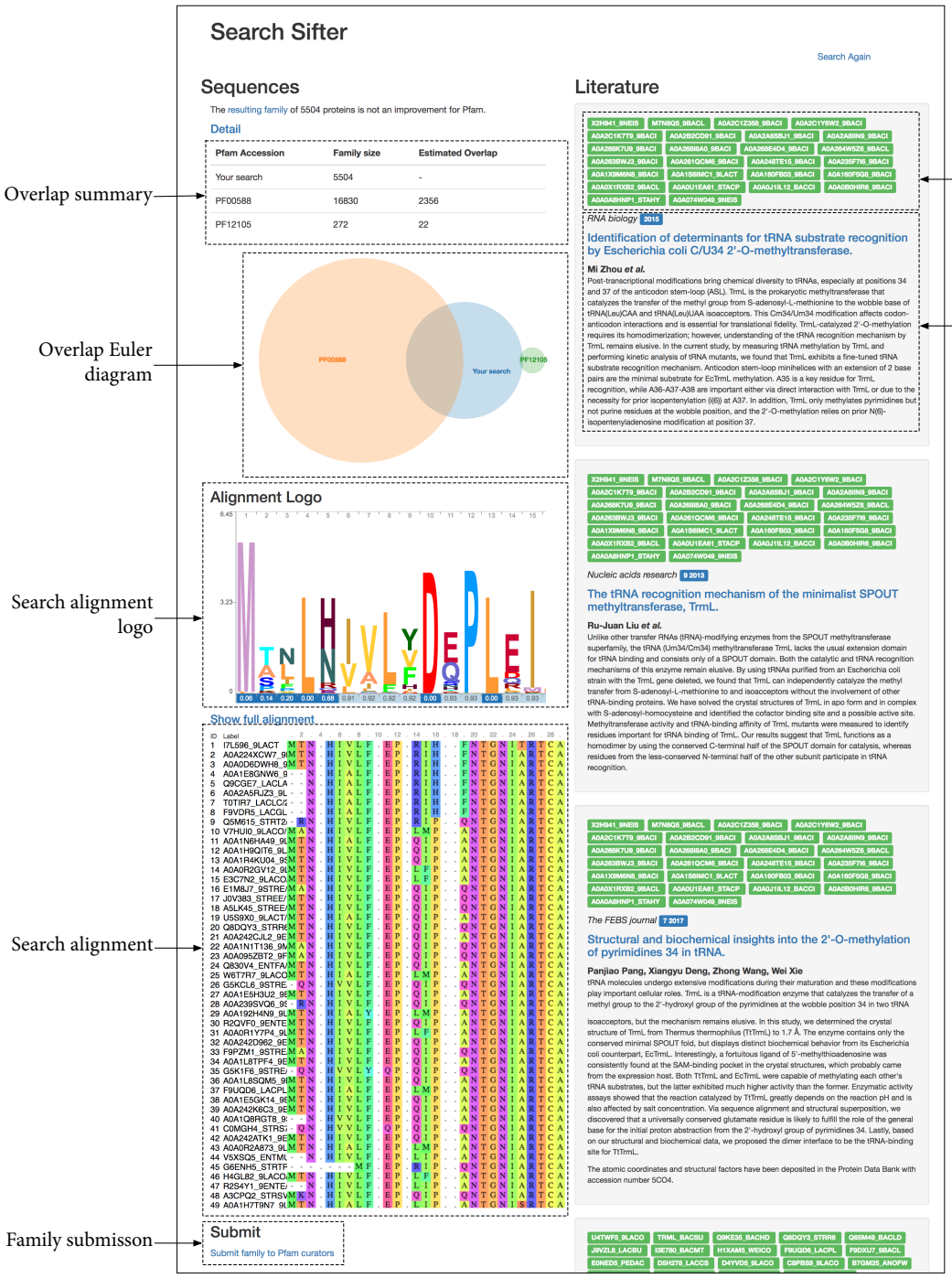


Figure 4.5. Screen capture of the Search-Sifter results user interface, for a search which with the SpoU rRNA Methylase (PF00588) and SpoU rRNA methylase C-terminal (PF12105) families. The screen capture is scaled down from its on-screen appearance in order to show all aspects of the user interface. Analysis of intersections of the search results and Pfam, and the search's alignment are shown on the left, and literature relevant to the search is shown on the right.

4.3. Conclusion

4.3.1. Use cases

Using the Search-Sifter web application, I identified two new protein families as a proof of principle, for inclusion in Pfam 32. The first is PF18050, which was named Cyclophilin-like family. The description is as follows: 'This entry represents a family of cyclophilin-like proteins found in a range of bacterial species.' The second is PF18701. This is a domain of unknown function (DUF), and was given the identifier DUF5641. A DUF is a protein family which contains only proteins of unknown function (Bateman *et al.*, 2010). The Search-Sifter user interface for the searches which identified these two families is shown in figure 4.7. The full Pfam descriptions are shown in figure 4.6. Their full alignments are given in section B.1.

```
ID   Cyclophil_like2
AC   PF18050
DE   Cyclophilin-like family
AU   Bateman A;0000-0002-6982-4660
SE   Jeffryes M
GA   22.70 22.70;
TC   22.70 22.70;
NC   22.60 22.60;
BM   hmmbuild  -o /dev/null HMM SEED
SM   hmmsearch -Z 45638612 -E 1000 --cpu 4
    HMM pfamseq
TP   Domain
CL   CL0475
CC   This entry represents a family of
    cyclophilin-like proteins found in
CC   a range of bacterial species.
```

a

```
ID   DUF5641
AC   PF18701
DE   Family of unknown function (DUF5641)
AU   Bateman A;0000-0002-6982-4660
AU   Jeffryes M;0000-0001-9868-6271
SE   Jeffryes M
GA   23.10 23.10;
TC   23.10 23.10;
NC   23.00 22.90;
BM   hmmbuild  -o /dev/null HMM SEED
SM   hmmsearch -Z 45638612 -E 1000 --cpu 4
    HMM pfamseq
TP   Domain
WK   Domain_of_unknown_function
CC   This presumed domain is found in a
    range of retrotransposon
CC   polyproteins.
ED   A0A2B4RHS3.1/800-869; A0A2B4RHS3
    .1/800-842;
```

b

Figure 4.6. The descriptions of the two protein families, PF18701 (a) and PF18701 (b), discovered using Search-Sifter.

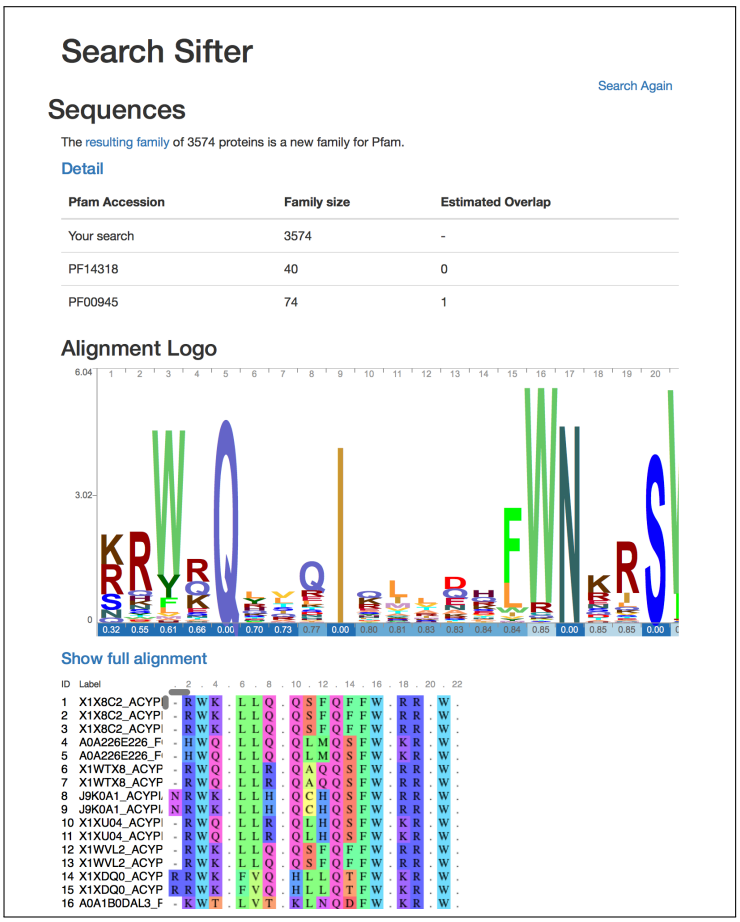


Figure 4.7. Screen capture of a recreation of the analysis of the HMMER search which identifies the new family PF18701, displayed in Search-Sifter. In the alignment listing, some accessions are displayed twice, due to domain repeats.

I anticipate that the Search-Sifter web application can be useful in several cases.

1. Users who believe they have identified a novel protein family using HMMER can check whether it overlaps with existing families.

2. Users who have conducted a HMMER search can identify literature relevant to the search results.

3. Curators can identify literature relevant to a protein family which they are in the process of annotating.

4.3.2. Future development

References to taxonomy are extracted from the text as part of the Grubbler literature processing pipeline. However, this information is not used. If this information was incorporated into the ranking of literature, more relevant results may be possible. A publication which mentions a protein found in the sequence similarity search results and the organism in which it is found may be more relevant than a publication which mentions the protein and an unrelated organism. Publications discussing more closely related orthologues may be more useful than literature discussing more distant orthologues. For example, if a family contains a Maize protein, a publication discussing the *Arabidopsis* orthologue may be more relevant than a publication discussing an animal orthologue.

Submission of novel families is possible using a pre-filled email, containing a link to the Search-Sifter results page for the family, the size of the potential new family and the alignment of the family in Stockholm format. It would be more user friendly to allow submission directly in the web application using a form. It would also streamline the curation process if users were able to highlight relevant literature to curators when submitting a family.

Search-Sifter may be useful in the curation not just of protein families but of proteins themselves. UniProt curators often work on the entries of several protein homologues at the same time, since their entries may be similar or identical. A literature search tool which searches for these homologues simultaneously might help identify relevant literature.

One could also imagine that this tool could be integrated into literature resources like Europe PubMed Central to enable a kind of sequence similarity search based tool for identifying related literature. For example, a reader could have the option to perform a HMMER search using sequences discussed in the current publication they are reading, and then via Search-Sifter, literature related to those sequences could be identified. This could identify literature which discusses proteins which are similar in sequence and structure but where the publications are lexically divergent, since such publications wouldn't be identified by typical 'similar article' tools which are based upon the terms found in the publications (Lin and Wilbur, 2007).

Similar methods to those discussed in this chapter could be used to assist in curation of other kinds of database. Rfam is a database of RNA families (Kalvari *et al.*, 2018). Using a Search-Sifter like tool with the results from the Infernal RNA homology search tool might assist in the curation of RNA families (Nawrocki and Eddy, 2013).

Of course, the purpose of developing this prototype was to experiment with methods which could be used to supply new protein families to Pfam, as part of the HMMER web service. In order to achieve this, considerable work would be required to ensure the software is fast and stable enough to be included in a web application depended upon by thousands of users. The

literature aspect of the software would need to be considerably developed, since it is not capable of continuously ingesting new incoming literature from PubMed. To be used in production, the software would need to be integrated into existing literature pipelines. Since this would require the input of development resources, stakeholders at the EMBL-EBI would need to be convinced that the project is worthwhile and of value to users. Further user-centric research would be needed to demonstrate this. Given the lack of dedicated funds for the HMMER web server, it is uncertain if the work discussed in this chapter will be incorporated into the HMMER web server.

4.4. References

Bartsch, A., B. Bunk, I. Haddad, J. Klein, R. Münch, T. Johl, U. Kärst, L. Jänsch, D. Jahn and I. Retter. 'GeneReporter - Sequence based document retrieval and annotation.' In: *Bioinformatics* 27.7 (2011), pp. 1034–1035. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btr047. PUBMED: 21310745.

Bateman, A., P. Coggill and R. D. Finn. 'DUFs: families in search of function.' In: *Acta crystallographica. Section F, Structural biology and crystallization communications* 66.Pt 10 (Oct. 2010), pp. 1148–1152. ISSN: 1744-3091. DOI: 10.1107/S1744309110001685. PUBMED: 20944204.

Dieterich, G., U. Kärst, J. Wehland and L. Jansch. 'MineBlast: a literature presentation service supporting protein annotation by data mining of BLAST results'. In: *Bioinformatics* 21.16 (Aug. 2005), pp. 3450–3451. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti528. PUBMED: 15941742.

Finn, R. D., J. Clements and S. R. Eddy. 'HMMER web server: interactive sequence similarity searching.' In: *Nucleic acids research* 39.Web Server issue (July 2011), W29–W37. ISSN: 1362-4962. DOI: 10.1093/nar/gkr367. PUBMED: 21593126.

Gilchrist, M. J., M. B. Christensen, R. Harland, N. Pollet, J. C. Smith, N. Ueno and N. Papalopulu. 'Evading the annotation bottleneck: using sequence similarity to search non-sequence gene data.' In: *BMC bioinformatics* 9 (Oct. 2008), p. 442. ISSN: 1471-2105. DOI: 10.1186/1471-2105-9-442. PUBMED: 18928517.

Jaroszewski, L., L. Koska, M. Sedova and A. Godzik. 'PubServer: literature searches by homology.' In: *Nucleic acids research* 42.Web Server issue (July 2014), W430–W435. DOI: 10.1093/nar/gku450. PUBMED: 24957597.

Kalvari, I., J. Argasinska, N. Quinones-Olvera, E. P. Nawrocki, E. Rivas, S. R. Eddy, A. Bateman, R. D. Finn and A. I. Petrov. 'Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families'. In: *Nucleic Acids Research* 46.D1 (Jan. 2018), pp. D335–D342. ISSN: 0305-1048. DOI: 10.1093/nar/gkx1038. PUBMED: 29112718.

Lin, J. and W. J. Wilbur. 'PubMed related articles: a probabilistic topic-based model for content similarity.' In: *BMC bioinformatics* 8 (Oct. 2007), p. 423. ISSN: 1471-2105. DOI: 10.1186/1471-2105-8-423. PUBMED: 17971238.

Mitchell, A. L., A. Divoli, J. H. Kim, M. Hilario, I. Selimas and T. K. Attwood. 'METIS: Multiple extraction techniques for informative sentences'. In: *Bioinformatics* 21.22 (2005), pp. 4196–4197. ISSN: 13674803. DOI: 10.1093/bioinformatics/bti675. PUBMED: 16159915.

Mitchell, A. L., I. Selimas and T. K. Attwood. 'Minotaur: A Web-Based Annotator-Assistant Tool'. In: *International journal of systems biology and biomedical technologies* 1.1 (2012), pp. 1–10. ISSN: 2160-9586. DOI: 10.4018/ijsbbt.2012010101.

Nawrocki, E. P. and S. R. Eddy. 'Infernal 1.1: 100-fold faster RNA homology searches'. In: *Bioinformatics* 29.22 (Nov. 2013), pp. 2933–2935. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt509. PUBMED: 24008419.

Price, M. N. and A. P. Arkin. 'PaperBLAST: Text Mining Papers for Information about Homologs.' In: *mSystems* 2.4 (2017). ISSN: 2379-5077. DOI: 10.1128/mSystems.00039-17. PUBMED: 28845458.

5. Implementing literature search for protein family curation

The previous chapter discusses the application of methods for quickly identifying high quality novel protein families, and of methods for identifying literature relevant to protein family curation, in the form of a web application targeted at protein sequence similarity search users. In this chapter, I will describe the application of methods for identifying literature relevant to protein family curation, in the form of a web application targeted at professional protein family curators.

The application discussed in this chapter is an extension to an internal EMBL-EBI web application called Pronto, which is used by staff curators of the InterPro database, the details of which are described in the following sections. The extension to Pronto allows curators to perform literature searches based on entries in InterPro, or in any of InterPro's member databases.

5.1. Background

5.1.1. InterPro curation

The InterPro database integrates information from fourteen member databases which contain information on protein classification. Each entry in the InterPro database is based on either a single entry in a member database, or on several similar entries across multiple databases. An entry which represents a more specific classification, that is, a subset of the proteins in another entry, can be linked hierarchically to its parent. As part of the integration process, entries are assessed for quality. Low quality entries from member databases are not integrated into InterPro (Finn *et al.*, 2017; Mitchell *et al.*, 2015)

Entries are manually annotated by staff curators at EMBL-EBI. Each entry has a free-text description, referenced to literature. The annotation of an InterPro entry requires the synthesis of information in one or more member database entries, and from the biomedical literature, which is a time consuming process (Finn *et al.*, 2017). Entries are sometimes revisited by cur-

ators between releases. For example, when the family in the member database changes. In this case, literature on the the entry may be reviewed again. However, this is not carried out systematically, and many entries have not been revisited by curators in recent years. For example, the Tubby domain (IPR000007) last had a publication curated into its entry in May 2006. If literature search were easier, it might enable more updates to older InterPro entries.

5.1.2. Pfam curation

Entries in Pfam are occasionally revisited by curators. This will occur if the profile hidden Markov model (HMM) for the family has been adjusted due to additions of new sequences, or if a newly created entry overlaps with it. But it can also occur if new information is available about an entry. Certain entries in Pfam are classified as domains of unknown function (DUFs). This means that the entry contains only protein sequences of unknown function (Bateman *et al.*, 2010). Naturally, new research could identify a function for a protein which was previously uncharacterised. Therefore, the annotation of a DUF entry should be revisited periodically to incorporate new information from the literature. Pfam contains around 5,000 DUF entries, so the process of checking for new literature relevant to these entries could be made more efficient through automation.

5.1.3. Pronto

InterPro curators have used a variety of internal tools. Currently, a new tool called Pronto is under development. Pronto allows summary and comparison of entries in member databases, through a variety of methods. These are, the proteins in the entries which are matched by other entries, the taxa in which the proteins in the entries were found, UniProt annotations which are shared between proteins in the entries, and Gene Ontology terms which are shared between the genes coding for proteins in the entries. Pronto also displays private annotations which curators add to entries in order to assist in future curation.

InterPro curators and Pfam curators do not use specialised literature search tools to assist them. In the first stage of literature search, the curator may look for members of the family which are in the manually curated Swiss-Prot subset of UniProt. As described in chapter 3, Swiss-Prot entries have a list of publications which were used in their curation. These publications may be relevant to the family to which they belong, particularly publications which discuss the protein's function or structure. If this approach does not produce sufficient curatable literature, the curator will query a general purpose biomedical literature repository such as PubMed or Europe PMC with appropriate search terms, such as the names of proteins in the

family (Alex Bateman, Sara El-Gebali, Lorna Richardson, Amaia Sangrador, personal communication, 2018).

I hypothesised that literature search by a similar method to that implemented in Search-Sifter could assist in both both InterPro curation and ongoing Pfam curation. However, curators would be less likely to use a tool which was not integrated with their existing workflow. Therefore, I developed a literature search tool, integrated into the Pronto interface, and with some modifications specific to these curation use-cases, based on the feedback of users.

5.2. Method

The extension to Pronto is facilitated through an addition to the Grubbler service, discussed in the previous chapter. For use with Pronto, the Grubbler service performs literature search based on families fetched from the InterPro Oracle database instead of from HMMER protein sequence similarity searches. A cartoon of this architecture is shown in figure 5.1

The tool allows curators to choose between the function, structure and sequence classifiers developed in chapter 3, for ranking of the literature. A key feature of Pronto, the ability to compare multiple families across the member databases, is incorporated into the tool by allowing users to search for literature which mentions proteins in one or more different families. In this case, users can filter the literature found in order to show only publications which match proteins found in a subset of the families being compared, or literature which mentions proteins found in every family being compared. This could facilitate the curation of a new superfamily entry for several InterPro entries.

The tool displays the context of possible matches in the literature, which allows curators to quickly determine whether the matched publication is relevant. It is also possible to filter out matches for a particular term. This enables curators to filter out matches for proteins which are not helpful, or which are due to spurious gene/protein named entity recognition (NER). The user interface of the results page is shown in figure 5.2.

Development of the Pronto literature search tool was conducted in consultation with curators. In the first phase of development, I demonstrated an initial version of the tool, and asked for feedback. The ability to filter out particular terms was based on this feedback, as was the use of colour to indicator of match score, and the ability to change the number of results shown per page. I then integrated suggested improvements. Following this, I asked the curators to use the tool when they felt it might be helpful, and to give me further feedback.

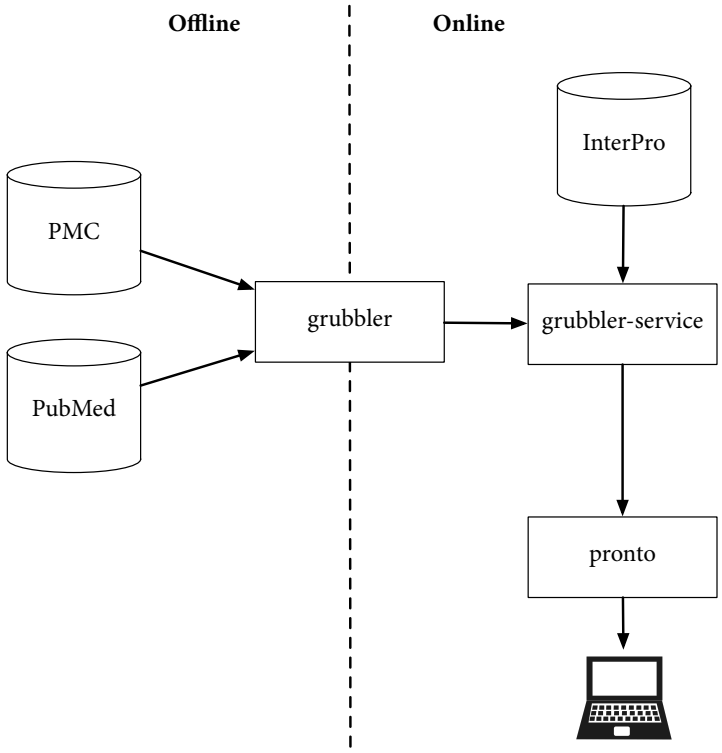


Figure 5.1. A cartoon of the Pronto literature search system. Literature is processed offline by Grubbler. When a search is performed, the members of the query family are fetched from the InterPro Oracle database.

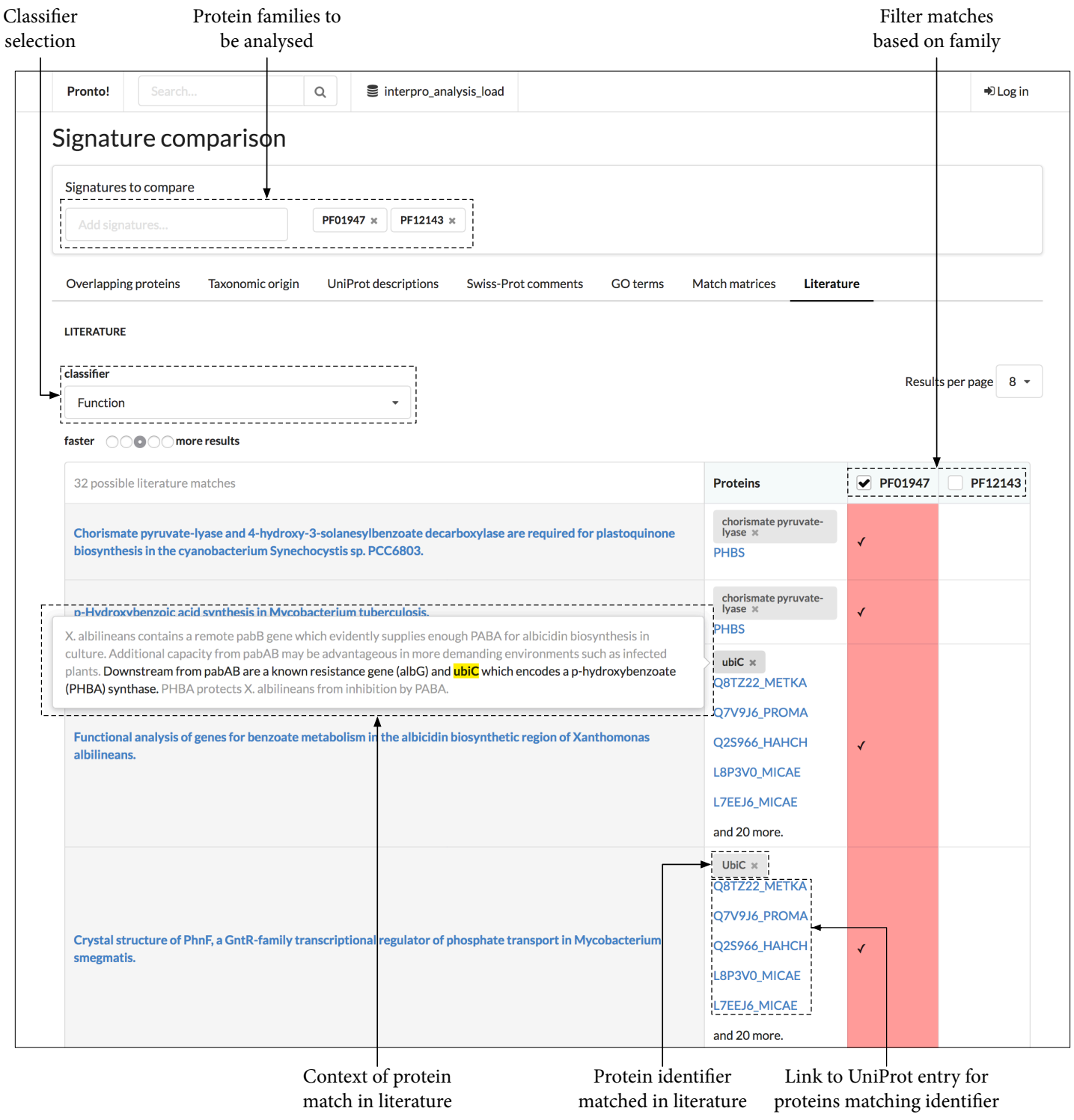


Figure 5.2. Screen capture of the Pronto literature search user interface. One or more families can be analysed. For each publication which mentions a protein in the families being analysed, the context of the match in the literature can be displayed by hovering on the protein name. The red background behind the tick for the families the matched protein appears in is faded for matches with a lower classifier score. The links to the left of the *Literature* label in the header area give access to the preexisting Pronto features.

5.3. Results

Evaluating the effectiveness of the system proved challenging. The curation staff have a high workload, and so an intensive assessment of their interaction with the system was not feasible. Instead, I solicited for feedback on the search results page, through the use of a 'thumbs up' and 'thumbs down' button, linking to a pre-filled Google Forms form, with space for the curator to enter more detailed feedback. The format of the feedback form and the responses are depicted in figure 5.3.

Of the four curators in the protein sequence family team, two used and gave feedback on the tool. One of the respondents was the sole full-time curator of Pfam, and the other was an InterPro curator. I received 17 and 7 responses respectively. In 18 (75%) of the searches for which the curator completed a survey, they believed that there were relevant results. Out of the 18 cases in which relevant results were found, in 13 cases the curator used the results to curate the family (54.2%). Of these 13 cases, in 6 cases, the curator believed that they could have found the literature curated without the literature search tool, but it would have taken longer, in 3 cases, they believed that they would not otherwise have found the literature, and in 2 cases, they believed that they would otherwise have found the literature faster without the tool, and in 2 cases, they were unsure.

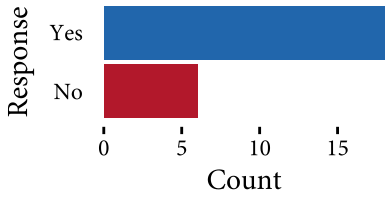
It's important to note that completing the survey was not mandatory, and the results are based purely on the curators' qualitative impressions of the tool.

5.4. Conclusion

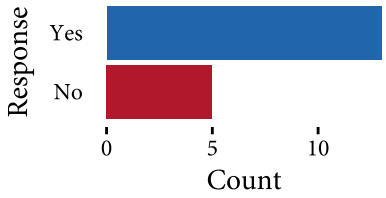
The assessment of the tool was limited to a qualitative and non-systematic pilot survey. While the results were encouraging, it cannot be stated that the tool is certainly an improvement over the curators' existing workflows. However, it is undeniable that the tool, in some cases, is able to produce useful literature search results.

There are several facets to the problem of characterising the effectiveness of the new system. There is no single obvious way of measuring whether the system improves curation 'performance'. One could imagine that a system which allows a curator to curate a new entry faster would be an improvement over a baseline. However, suppose that the baseline system finds only a single relevant publication, whereas the new system finds two relevant publications. In this case, the new system may cause the curation of the entry to take longer, but the entry created will be more complete. On the other hand, using a larger number of publications in the curation of a database entry is not necessarily better. Suppose that the baseline system, the

1. Were any of the results relevant? *24 responses*



2. Did any of the results assist you in curating the family? *18 responses*



3. Which results helped you? *13 responses*

Free text

4. Do you think you would have found this publication without the literature search tool?

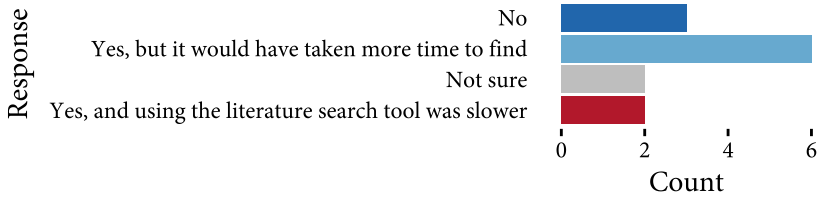


Figure 5.3. The format of and responses to the Pronto literature search tool survey. Only the first question was answered in all cases. The remaining questions were answered depending on whether they were appropriate (a search which returns no relevant results cannot help in curating a family).

curator identifies two relevant publications, and curates them into the new entry, whereas the new system identifies these two publications, and an additional publication which includes all the information found in the other two publications. In this case, using the new system, the curator may curate this single publication into the new entry, and will not require the other two publications.

In order to find further relevant literature, the tool could be extended to use other methods. Since Swiss-Prot references are a frequent source of relevant literature for InterPro and Pfam curators, it would be useful to display references for Swiss-Prot proteins within the family, and this could be filtered or sorted using the scope annotation of the reference.

The method for literature search implemented in the extension to Pronto is a promising enhancement to the curator workflow, and by incorporating other methods of literature search, could replace the manual search currently performed by InterPro curators.

Since the tool discussed in this chapter uses the same literature search service developed in the previous chapter, the same requirements for deploying it more widely apply: The literature service would require development so that it is able to continuously ingest new literature. This would require significant software development to make the software stable and scalable. As discussed in the previous chapter, a source of funding would be needed to make these developments.

5.5. References

Bateman, A., P. Coggill and R. D. Finn. 'DUFs: families in search of function.' In: *Acta crystallographica. Section F, Structural biology and crystallization communications* 66.Pt 10 (Oct. 2010), pp. 1148–1152. ISSN: 1744-3091. DOI: 10.1107/S1744309110001685. PUBMED: 20944204.

Finn, R. D., T. K. Attwood, P. C. Babbitt, A. Bateman, P. Bork, A. J. Bridge, H.-Y. Chang, Z. Dosztányi, S. El-Gebali, M. Fraser, J. Gough, D. Haft, G. L. Holliday, H. Huang, X. Huang, I. Letunic, R. Lopez, S. Lu, A. Marchler-Bauer, H. Mi, J. Mistry, D. A. Natale, M. Necci, G. Nuka, C. A. Orengo, Y. Park, S. Pesseat, D. Piovesan, S. C. Potter, N. D. Rawlings, N. Redaschi, L. Richardson, C. Rivoire, A. Sangrador-Vegas, C. Sigrist, I. Sillitoe, B. Smithers, S. Squizzato, G. Sutton, N. Thanki, P. D. Thomas, S. C. E. Tosatto, C. H. Wu, I. Xenarios, L.-S. Yeh, S.-Y. Young and A. L. Mitchell. 'InterPro in 2017—beyond protein family and domain annotations.' In: *Nucleic acids research* 45.D1 (Jan. 2017), pp. D190–D199. ISSN: 0305-1048. DOI: 10.1093/nar/gkw1107. PUBMED: 27899635.

Mitchell, A., H.-Y. Chang, L. Daugherty, M. Fraser, S. Hunter, R. Lopez, C. McAnulla, C. McMenamin, G. Nuka, S. Pesseat, A. Sangrador-Vegas, M. Scheremetjew, C. Rato, S.-Y. Yong, A.

Bateman, M. Punta, T. K. Attwood, C. J. A. Sigrist, N. Redaschi, C. Rivoire, I. Xenarios, D. Kahn, D. Guyot, P. Bork, I. Letunic, J. Gough, M. Oates, D. Haft, H. Huang, D. A. Natale, C. H. Wu, C. Orengo, I. Sillitoe, H. Mi, P. D. Thomas and R. D. Finn. 'The InterPro protein families database: the classification resource after 15 years.' In: *Nucleic acids research* 43.Database issue (Jan. 2015), pp. D213–D221. ISSN: 1362-4962. DOI: 10.1093/nar/gku1243. PUBMED: 25428371.

6. Conclusion

The rapid increase in biological data, from the late twentieth century onwards, has been facilitated by new scientific and engineering developments, primarily in the area of nucleotide sequencing, but secondarily by developments in computer processor, network and storage technology. As the cost of storage has shrunk, so too has the cost of genomic and proteomic data acquisition (Sboner *et al.*, 2011; Stephens *et al.*, 2015). The operators of biological archives, such as the European Nucleotide Archive (ENA), are faced with the challenge of ensuring that the size of their collection does not exceed their ability to pay for it, spurring innovation in areas such as data compression (Cochrane *et al.*, 2012).

In contrast, the operators of a curated database are not limited by storage or compression technology, but by the high cost of human labour. Protein families are invariably manually curated (see chapter 1). The curator of such a database is faced not with a shortage of data, nor with a lack of hard drive space, but with the limit of their own time. Thus, to advance the goal of identifying as many protein families as possible, we require developments which will make curators more efficient, and able to curate more database entries in their limited and valuable time.

6.1. Discussion

The initial goal of the project described in this thesis was to facilitate crowdsourcing of the Pfam database. Through crowdsourcing, we can offload some of the labour of curating Pfam on to the community of protein family database users. Users already take the time to construct alignments and submit them by email to the Pfam curators. But I believe that through new methods and tools, more users can be recruited into submitting families, and they can perform a larger proportion of the work of curation. I have described several methods which will enable this, and developed tools which demonstrate the use of these methods.

In the first chapter, I discuss my investigation into the quality of alignments produced by HMMER searches, and methods for measuring it. I also described methods for rapidly comparing HMMER search results to the existing families in Pfam. The first outcome of the work in this

chapter was the validation of the use of a reversed sequence database to identify low quality alignments which would rapidly expand in size. This method could be applied as a filter to the submission of crowdsourced families to Pfam. I believe that this technique could have further applications beyond this. It could be incorporated into the HMMER web server, and be used in the calculation of a quality measure for HMMER searches, being particularly useful when the user performs *jackhmmer* searches.

The second outcome of this work was the application of locality sensitive hashing to the comparison of protein families. Using this method, a protein family can be compared with Pfam in less than a second, to identify whether it is a novel grouping, or an improvement over an entry already in Pfam. In principle, this same method could be used for comparing groupings in other kinds of sequence data, for example, RNA families.

In the second chapter, I discuss the use of literature search to facilitate crowdsourced curation. In particular, that curation is not simply identifying a novel family, but annotating further information sourced from the literature. I used the literature curated in the Swiss-Prot subset of the UniProt database to train classifiers which identify curatable literature on proteins. I developed the Grubbler literature processing system to process literature, extract features, and train and experiment with models, and a web service to retrieve bibliographic information and classifier scores.

In the third chapter, I applied the methods developed in the previous two chapters to develop the Search-Sifter web application, which analyses HMMER web searches by comparing them to the existing families in Pfam, and determines whether they could be included in Pfam either as novel families or as improved models for existing families.

During the work of this chapter, I identified that the same techniques that could be used to make curation by the community more expansive could also be used to make curation by staff curators more efficient.

In the fourth chapter, I adapted the literature search methods developed in the second chapter to the needs of professional curators of the InterPro database. I integrated and enhanced the previously developed Grubbler web service with a curation tool, Pronto, to allow curators to perform literature searches using InterPro entries.

While some of the work discussed in this thesis shows promise, I was unable to realise the goal of integrating crowdsourced curation into the Pfam database. I was able to develop methods and proofs of concept, but the significant development resources required to integrate these with the existing EMBL-EBI web service infrastructure and the rigorous testing that would be required in order to ensure that the availability of existing, widely used web services was main-

tained would have required an investment of time and resources only possible if the project were seen as worthwhile in the context of the broader goals of the EMBL-EBI service offering.

The initial goal of integrating the tool with HMMER then, may have been too ambitious. It may have been better to aim to produce a standalone tool for submitting to Pfam, as this would have not required the costly and complex integration with the existing HMMER web service.

Another deficiency was the scale of testing for the literature search system, Grubbler. In order to show that such a tool improves curation, it is necessary to perform extrinsic testing, and to measure whether integrating the tool into a curation workflow actually improves the curation product, either in the quality of entries, or the rate at which they are curated. Such testing would require considerable work from curators. The evaluation of the curation tool, PubTator required two curators to curate 25 publications each (Wei *et al.*, 2012). Each of these 50 publications had previously been curated as a gold standard by other curators. This kind of formal testing, even over what might appear to be a small data set requires a significant time commitment, again necessitating justification of the importance of the service. This is why testing with curators in this project was limited to an informal survey, but of course this reduces the persuasiveness of the work.

6.2. Future work

During this project, I have yet to develop the methods and tools discussed into a production web application. Ideally, the first chapter work would be integrated into the HMMER web server, allowing users of the protein sequence search to submit their searches directly to Pfam. As part of this, literature search could also be integrated, both as a means for users to create families which they submit, and as a useful tool by itself. This would bring the literature search demonstrated in Search-Sifter to a wider audience.

As noted previously, the testing of the literature search tools was not sufficiently rigorous. For the literature search tool integrated into Pronto, testing could involve curators repeating the curation for existing Pfam entries, with one curator using their normal workflow and another curator using the Pronto literature search tool. Evaluation could involve comparing the set of references included in the entry produced by each entry, the time taken to produce each entry, and a blind evaluation by a third curator of the quality of each entry. As noted previously, this would be a considerable resource investment.

One aspect of the project which was not at all evaluated was whether crowdsourced annotations for new families would be high quality. Based on a model where crowdsourced families are submitted to a curator for approval, with or without some modification, it would be essen-

tial to determine whether this process saves time and results in equal or higher quality families. Such testing would involve a curator re-curating a crowdsourced family, and again comparing the references included in the two entries, the time taken to curate the entry compared to the time taken to process a crowdsourced entry, and a blind evaluation of quality. It's obviously an unanswered question whether it is worthwhile asking users to assist in the annotation process, as opposed to just submitting possible new families.

Once users are able to submit new families, the question arises as to whether they need to be encouraged to do so, and how this might be achieved. A strategy within crowdsourcing to increase participation is to reframe the task as a game (Good and Su, 2013). Rather than receiving a material reward, the participant is rewarded with entertainment. For example, Baker and colleagues developed Foldit, a game which challenges players to solve protein structure problems (Cooper *et al.*, 2010). Foldit players are able to achieve high performance in constructing structures, outperforming automated methods and trained crystallographers (Horowitz *et al.*, 2016).

It may not be feasible or desirable to convert all tasks that you might want to crowdsource into a game. Gamification techniques include the addition of elements normally found in video games to other tasks. For example, assigning 'levels' to a user, awarding virtual 'badges' or 'trophies', and comparing the progress of different users on a leaderboard (Dubois and Tamburrelli, 2013). While the task itself may not be regarded as fun, the feeling of accomplishment from completing it can be enhanced by adding mechanisms which track and recognise progress. For example, peer review is often regarded as a tiresome chore (Grainger, 2007). The Publons service gamifies some aspects of it, by allowing researchers to track how many reviews they have completed, publicly identifying researchers who contribute many reviews, and awarding virtual prizes (Ravindran, 2016).

Crowdsourced protein family curation could be gamified in a similar way. Users could compete to create families which add the most coverage to the Pfam database, along with other tasks, like curating new literature to existing families or identifying the function of domain of unknown function (DUF) families.

Gamification could lead to participants developing novel ways for identifying and curating new families, in their attempt to outperform other participants. Cooper *et al.* (2010) note that 'Humans use a much more varied range of exploration methods than computers. Different players use different move sequences, both according to the puzzle type and throughout the duration of a puzzle', and that rather than exploring only the search space they explore 'the space of possible search strategies'. It is conceivable that a creative participant could take a novel approach to the task of family identification, enabling them to find areas of proteins space yet

unclassified, or that they could take a novel approach to literature curation, allowing them to mine information from the literature which is not yet present in Pfam. Since Pfam is based upon open access data, it would be feasible for participants to develop their own, novel automated methods.

My contributions are to demonstrate the feasibility of using protein sequence similarity searches to crowdsource the identification of protein families, and to use these searches as the basis for literature search, enabling the crowdsourced curation of these families. I have developed prototypes which have promising utility, but the task of integrating these methods into a production workflow is left for the future.

The identification and annotation of protein families and domains is an important task. As we enter the third decade of the genomic era, the interpretation of sequence data becomes more and more pervasive a task in biology. Databases like Pfam and UniProt connect biologists to literature. I hope that my work can help in some way to ensure they see literature which facilitates their understanding.

6.3. References

Cochrane, G., B. Alako, C. Amid, L. Bower, A. Cerdeño-Tárraga, I. Cleland, R. Gibson, N. Goodgame, M. Jang, S. Kay, R. Leinonen, X. Lin, R. Lopez, H. McWilliam, A. Oisel, N. Pakseresht, S. Pallreddy, Y. Park, S. Plaister, R. Radhakrishnan, S. Rivière, M. Rossello, A. Senf, N. Silvester, D. Smirnov, P. ten Hoopen, A. Toribio, D. Vaughan and V. Zalunin. 'Facing growth in the European Nucleotide Archive'. In: *Nucleic acids research* 41.D1 (Nov. 2012), pp. D30–D35. ISSN: 0305-1048. DOI: 10.1093/nar/gks1175. PUBMED: 23203883.

Cooper, S., F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker and Z. Popović. 'Predicting protein structures with a multiplayer online game.' In: *Nature* 466.7307 (Aug. 2010), pp. 756–60. DOI: 10.1038/nature09304. PUBMED: 20686574.

Dubois, D. J. and G. Tamburrelli. 'Understanding gamification mechanisms for software development'. In: *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2013*. New York, New York, USA: ACM Press, 2013, p. 659. ISBN: 9781450322379. DOI: 10.1145/2491411.2494589.

Good, B. M. and A. I. Su. 'Crowdsourcing for bioinformatics.' In: *Bioinformatics* 29.16 (Aug. 2013), pp. 1925–1933. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt333. PUBMED: 23782614. arχiv: 1302.6667.

Grainger, D. W. 'Peer review as professional responsibility: A quality control system only as good as the participants'. In: *Biomaterials* 28.34 (Dec. 2007), pp. 5199–5203. ISSN: 0142-9612. DOI: 10.1016/J.BIOMATERIALS.2007.07.004.

Horowitz, S., B. Koepnick, R. Martin, A. Tymieniecki, A. A. Winburn, S. Cooper, J. Flatten, D. S. Rogawski, N. M. Koropatkin, T. T. Hailu, N. Jain, P. Koldewey, L. S. Ahlstrom, M. R. Chapman, A. P. Sikkema, M. A. Skiba, F. P. Maloney, F. R. M. Beinlich, Foldit Players, University of Michigan students, Z. Popović, D. Baker, F. Khatib and J. C. A. Bardwell. 'Determining crystal structures through crowdsourcing and coursework'. In: *Nature Communications* 7.1 (Dec. 2016), p. 12549. ISSN: 2041-1723. DOI: 10.1038/ncomms12549.

Ravindran, S. 'Getting credit for peer review'. In: *Science Careers* (Feb. 2016). ISSN: 10959203. DOI: 10.1126/science.caredit.a1600022.

Sboner, A., X. J. Mu, D. Greenbaum, R. K. Auerbach and M. B. Gerstein. 'The real cost of sequencing: higher than you think!' In: *Genome biology* 12.8 (Aug. 2011), p. 125. ISSN: 1474-760X. DOI: 10.1186/gb-2011-12-8-125. PUBMED: 21867570.

Stephens, Z. D., S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha and G. E. Robinson. 'Big Data: Astronomical or Genomical?' In: *PLoS biology* 13.7 (July 2015), e1002195. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1002195. PUBMED: 26151137.

Wei, C.-H., B. R. Harris, D. Li, T. Z. Berardini, E. Huala, H.-Y. Kao and Z. Lu. 'Accelerating literature curation with text-mining tools: a case study of using PubTator to curate genes in PubMed abstracts'. In: *Database* 2012 (Nov. 2012), bas041. ISSN: 1758-0463. DOI: 10.1093/database/bas041. PUBMED: 23160414.

List of Publications

M. Jeffryes and A. Bateman. 'Rapid identification of novel protein families using similarity searches'. In: *F1000Research* 7 (Dec. 2018), p. 1975. ISSN: 2046-1402. DOI: 10.12688/f1000research.17315.1

W. Höps, M. Jeffryes and A. Bateman. 'Gene Unprediction with Spurio: A tool to identify spurious protein sequences'. In: *F1000Research* 7 (Mar. 2018), p. 261. ISSN: 2046-1402. DOI: 10.12688/f1000research.14050.1

A. Prakash, M. Jeffryes, A. Bateman and R. D. Finn. 'The HMMER Web Server for Protein Sequence Similarity Search'. In: *Current Protocols in Bioinformatics* 60.1 (Dec. 2017), pp. 3.15.1–3.15.23. ISSN: 19343396. DOI: 10.1002/cpbi.40. PUBMED: 29220076

Appendices

A. Literature

A.1. Scope categories

On the left, fragments of text from the scope annotations of references for Swiss-Prot entries, and on the right, the corresponding category with which the entry should be labelled. These were provided by Cecilia Arighi (personal communication, 2017).

Scope	Category
NUCLEOTIDE SEQUENCE	Sequences
FUNCTION	Function
IDENTIFICATION	Unclassified
SUBCELLULAR LOCATION	Subcellular location
INTERACTION	Interaction
PHOSPHORYLATION	PTM/processing
TISSUE SPECIFICITY	Expression
PROTEIN SEQUENCE	Sequences
GENOME REANNOTATION	Sequences
X-RAY CRYSTALLOGRAPHY	Structure
VARIANT	Sequences
MUTAGENESIS	Pathology & Biotech
INDUCTION	Expression
SUBUNIT	Interaction
DISRUPTION PHENOTYPE	Pathology & Biotech
VARIANTS	Sequences
GENE FAMILY	Names
CATALYTIC ACTIVITY	Function
NOMENCLATURE	Unclassified
DEVELOPMENTAL STAGE	Expression
CHARACTERIZATION	Unclassified

Scope	Category
ACETYLATION	PTM/processing
BIOPHYSICOCHEMICAL PROPERTIES	Function
STRUCTURE	Structure
GLYCOSYLATION	PTM/processing
ENZYME REGULATION	Function
REVIEW	Unclassified
SEQUENCE REVISION	Sequences
MASS SPECTROMETRY	Sequences
COFACTOR	Function
CLEAVAGE	PTM/processing
ALTERNATIVE SPLICING	Sequences
LEVEL OF TISSUE EXPRESSION	Expression
INVOLVEMENT	Pathology & Biotech
DOMAIN	Family & Domains
DISULFIDE BONDS	PTM/processing
TOPOLOGY	Subcellular location
UBIQUITINATION	PTM/processing
AMIDATION	PTM/processing
SUBSTRATE SPECIFICITY	Function
DNA-BINDING	Function
SUMOYLATION	PTM/processing
PATHWAY	Function
ACTIVE SITE	Function
DISULFIDE BOND	PTM/processing
3D-STRUCTURE MODELING	Structure
METHYLATION	PTM/processing
GENE FAMILY ORGANIZATION	Family & Domains
SYNTHESIS	Unclassified
RNA-BINDING	Function
PROTEOLYTIC PROCESSING	PTM/processing
GENE NAME	Names
RNA EDITING	Sequences
ASSOCIATION	Unclassified
ROLE	Function

Scope	Category
POLYMORPHISM	Sequences
AUTOPHOSPHORYLATION	PTM/processing
REACTION MECHANISM	Function
ENZYME ACTIVITY	Function
PYROGLUTAMATE FORMATION	PTM/processing
CHROMOSOMAL TRANSLOCATION	Pathology & Biotech
GENE MODEL	Sequences
SUCCINYLATION	PTM/processing
EXPRESSION	Expression
REGULATION	Function
PALMITOYLATION	PTM/processing
LETHAL DOSE	Pathology & Biotech
HYDROXYLATION	PTM/processing
CRYSTALLIZATION	Structure
HOMODIMERIZATION	Interaction
GPI-ANCHOR	PTM/processing
KINETIC PARAMETERS	Function
MYRISTOYLATION	PTM/processing
ELECTRON MICROSCOPY	Structure
BIOTECHNOLOGY	Pathology & Biotech
DISEASE	Pathology & Biotech
COMPONENT	Unclassified
OPERON STRUCTURE	Unclassified
SIMILARITY	Family & Domains
ALLERGEN	Pathology & Biotech
DEPHOSPHORYLATION	PTM/processing
INHIBITION	Function
FUNCTION (MICROBIAL INFECTION)	Function
COMPOSITION	Sequences
BINDING	Function
DOMAINS	Family & Domains
ALTERNATIVE INITIATION	Sequences
CALCIUM-BINDING	Function
SULFATION	PTM/processing

Scope	Category
SEQUENCE	Sequences
PROTEOLYTIC CLEAVAGE	PTM/processing
NUCLEAR LOCALIZATION SIGNAL	Family & Domains
REGION	Family & Domains
SELF-ASSOCIATION	Interaction
ISOPRENYLATION	PTM/processing
MUTANT	Pathology & Biotech
TRANSCRIPTIONAL REGULATION	Function
COMPLEX	Interaction
ACTIVE SITES	Function
ZINC-BINDING	Function
DIMERIZATION	Interaction
ALTERNATIVE PROMOTER USAGE	Sequences
DORMANCY REGULON	Unclassified
MISCELLANEOUS	Unclassified
EPR SPECTROSCOPY	Structure
OLIGOMERIZATION	Interaction
ACTIVATION	Function
PUPYLATION	PTM/processing
GAMMA-CARBOXYGLUTAMATION	PTM/processing
DEUBIQUITINATION	PTM/processing
RECONSTITUTION	Sequences
DEGRADATION	PTM/processing
COMPLETE GENOME	Sequences
SPLICE ISOFORM(S) THAT ARE POTENTIAL NMD TARGET(S)	Sequences
MUTANTS	Pathology & Biotech
AUTOUBIQUITINATION	PTM/processing
S-NITROSYLATION	PTM/processing
ZINC-BINDING SITES	Function
BLOCKAGE	PTM/processing
BIOASSAY	Unclassified
AUTOCATALYTIC CLEAVAGE	PTM/processing
MEMBRANE TOPOLOGY	Subcellular location
ATP-BINDING	Function

Scope	Category
CITRULLINATION	PTM/processing
TOXIN TARGET	Pathology & Biotech
REPRESSION	Function
PTM	PTM/processing
OXIDATION	PTM/processing
CROSS-LINKING	PTM/processing
HETERODIMERIZATION	Interaction
TOXIC DOSE	Pathology & Biotech
CIRCULAR DICHROISM ANALYSIS	Structure
OVEREXPRESSION	Unclassified
COMPLETE PLASTID GENOME	Sequences
METAL-BINDING SITES	Function
LIPID-BINDING	Function
ISGYLATION	PTM/processing
DEACETYLATION	PTM/processing
CLEAVAGE SITE	PTM/processing
NUCLEAR EXPORT SIGNAL	Family & Domains
CALCIUM-BINDING SITES	Function
ATPASE ACTIVITY	Function
SUBSTRATES	Function
PYRIDOXAL PHOSPHATE	PTM/processing
HEME-BINDING	Function
DEAMIDATION	PTM/processing
LIGAND-BINDING	Function
HOMOOLIGOMERIZATION	Interaction
PH DEPENDENCE	Function
FORMYLATION	PTM/processing
GENE STRUCTURE	Sequences
CIRCULAR DICHROISM	Structure
CANDIDATE MALARIA EPITOPE	Unclassified
CROTONYLATION	PTM/processing
EFFECT	Unclassified
CARBAMYLATION	PTM/processing
ADP-RIBOSYLATION	Function

Scope	Category
GTP-BINDING	Function
NITRATION	PTM/processing
HOMODIMER	Interaction
CONCEPTUAL TRANSLATION	Sequences
CONJUGATION	PTM/processing
COILED-COIL DOMAIN	Family & Domains
PARALYTIC DOSE	Pathology & Biotech
GENE FAMILY AND NOMENCLATURE	Names
UBIQUITIN-BINDING	Interaction
PRESENCE	Unclassified
MECHANISM	Function
CATALYTIC MECHANISM	Function
ANTIBIOTIC RESISTANCE	Unclassified
RIBOSOMAL FRAMESHIFT	Sequences
MOTIF	Family & Domains
METAL-BINDING	Function
D-AMINO ACID	PTM/processing
SUBSTRATE-BINDING SITES	Function
SIGNAL SEQUENCE CLEAVAGE SITE	PTM/processing
PHOSPHOPANTETHEINYLATION	PTM/processing
DIACYLGLYCEROL	PTM/processing
GENE DUPLICATION	Sequences
AMINO-ACID COMPOSITION	Sequences
METHYLTRANSFERASE ACTIVITY	Function
CHROMOSOMAL REARRANGEMENT	Sequences
PROTEASOMAL DEGRADATION	PTM/processing
SELENOCYSTEINE	Sequences
ACTIVITY PROFILE	Function
SITE	Function
POST-TRANSLATIONAL MODIFICATIONS	PTM/processing
HETERODIMER	Interaction
GLYCYLATION	PTM/processing
COMPLEX FORMATION	Interaction
STOICHIOMETRY	Interaction

Scope	Category
PROCESSING	PTM/processing
IMPRINTING	Unclassified
DOWN-REGULATION	Unclassified
REPEATS	Family & Domains
INTERCHAIN DISULFIDE BOND	PTM/processing
IMPORTANCE	Unclassified
ASSEMBLY	Unclassified
TRANSIT PEPTIDE CLEAVAGE SITE	PTM/processing
BIOSYNTHESIS	Function
SECRETION VIA TYPE III SECRETION SYSTEM	Subcellular location
REQUIREMENT	Unclassified
MODELING	Structure
COPPER-BINDING	Function
ACTIN-BINDING	Function
NPC SUBUNIT LOCATION	Unclassified
GLYCATION	PTM/processing
ENZYMATIC ACTIVITY	Function
CAUTION	Unclassified
RETRACTION	Unclassified
NEDDYLATION	PTM/processing
IRON-BINDING SITES	Function
HEPARIN-BINDING	Function
FORMATION	Unclassified
CROSS-LINK	PTM/processing

B. Search-Sifter

B.1. Pfam families discovered using Search-Sifter

The seed alignments for the two Pfam families discovered using Search-Sifter, which are incorporated into Pfam 29.

B.1.1. PF18050

```
A0A133SGG3.1/30-141     IYFGD..ILVPATL.NDSKAA.KALIKML...PYK....VRVNRYS....FDVCGV..MGN.ALPYDPK.DEHNGWLN..GDIDFATDGNWFTILFDNEENSDSYG.YQVNLGK.V........DSELSVL...KNLSGSY...EVRIE
C6BTD0.1/175-282        VIANG..HEIVFEL.NDSQAS.KDLYAQL...PMR....ISVDNYG.S..NEKIFY..PAK.KLGTSNT.PLVKSATV..GTLAYYAPWGDVVMFYGSF..GS.AS.GLYELGK.A...IQ..GSEH...I...RKLSG.....TIKIE
A0A0D6WPP5.1/21-128     VTLDG..RPVGATL.HDTPPA.RDFAKLL...PLT....LELEDFH.D..TERVAA..LPR.RLDTSGA.PEPVRAGP..GDIAYYAPWGNLALFHRDG..PAPSA.DLLVLGR.L...DA..GPGQ.........LGRAT...RITLE
A0A0R2BRX4.1/25-136     LIINN..QTFPAIL.NDSQAT.QTLVKKL...PYT....ITVIQEI....HDYCGV..MDH..LPYSKD.DVQVGWFD..GDLAFDISGDWFAFFLRGANNDVRYR.E.VNLGQ.L......VNKEEIARI...AQLPATV...DITIE
E7G804.1/6-114          MIINN..QEWQVNL.YQNDTV.ESLINRL...PLT....IQMKELH.G..NEKYHY..LDF.QLPIMSE..SIDQIET..GDLMLFGN.NCLVLFYESFST...PY.QYTKIGY.I.....DNLKNIKNLV...GS..KDV...KVTFQ
A0A0A1CSU6.1/5-114      LSIDG..IHVEAEL.YDHPVA.GELAGML...PLD....LIFNDFN.N..VEKVAS..LGR.SLTLHGV.PDADAPQP..GEIGYYAPTQGFVLFYGSPGR...WP.GLVRMGR.F...SY..DLEA...L...RDLPDAT...SIHIA
Q46EF0.1/95-202          ITSQG..HIATFQL.YDTVAS.KELYEQL...PLE....LDLTNFR.D..AQWMFY...PPKKLNVTAQ.EAYHDGKK..GELSYYGPWGDVFMLYEDFYA...GD.EMHRLGICITGINEIANMSG.................SMQIE
Q9PCM7.1/55-163         LIVGE..QVATATL.YDNATA.KDFASLL...PLS....LTMTDYD.T..IERVSD..LPR.KLSTQGA.PEGVAPVA..GELTHYAPWGNLAIFIKPR..SY.SR.SLLPLGK.I...DD..GLAI...V...S.QPGPY...KMRIE
E2SLJ8.1/29-137          MVMNN..QTFSATL.QDNETV.RALIKQM...PMT....LDMADLH.G..NEKYHY..FSN.GFPGKAQ..PVAQILT..GEIKLFGD.DCLVVFYKDFTT...TY.SYISLGR.V.....DDPKAFAKAM...NK..GNV...QVRFE
C0BXI1.1/65-174          LTFEG..GEAMVRL.NDNAAA.QSFAAQL...PMT....QTFEDFN.S..IEKICR..LQE.ELTTEGV.ESGVDPAV..ADITLYVPWNTLVFYYEDY..GF.ND.DLIPMGR.V...ES..GMEL...L...TAMGDEF...EVTMD
D5QDI5.1/5-113           LTLGE..QVIFASL.EDTPSG.RDFLSLV...SMT....LMLEDYN.A..TEKICD..LPR.HLSSDKA.PKGYAPSA..GDIAYYAPWGNIAIFYRNF..EY.SS.GLIKLGQ.I...ED..NLEG...L...A.LTGRL...KMQVE
A0A0B1YL45.1/5-113       ISIDG..RKVVAEL.EDSATS.RDFLSQL...PMT....LKFEDYH.A..TEKISY..LPR.KLSTSGA.PDGFDPSV..GTVAYYAPWGNIAVFYNDF..GY.SR.SLIKLGQ.V...VS..GLDI.......LTRASSF...NAKIE
U2DA56.1/50-156          VRADG..REIVYEL.GQSAAA.QGLCDQL...PLT....VEVEDFS.T..NEKIFY..PPQ.ELEVADA.PLA.EGGR..GVLAYYAPWGDVVLFYGPF..DG.SG.QLYELGR.A...VS..GEGD...I...EALSG.....SVTIS
R7FLK7.1/63-172          LTLDS..SDVDVYW.MDNDSV.KEL.KKLAKDGLT....IELHQYG.G..FEQVGS..LGS.TIKSNDS.....SITTNAGDICLYQS.NQIVFFYGSN.....TL.SYTKLGH.I....NLTKTELVELL...GEE.DTV...TITLN
A0A1B1KRF0.1/44-153      IAIGS..TYLTATL.EDNPTA.RSFVKLL...PLT....ITLKDYG.D..AEKISG.ALPH.SLSQDAA.PESAAGAT..GDIAYYAPWGNIAF.YRGR..GPEAA.GVIKIGK.I...TS..GIEA.......LNQPGSM...RVTIA
D1Y1E5.1/29-140          ILIDG..KTYAAAL.EDNVTA.RDIAARL...PLE....LDMKRFG.G..HEFYAE..LP...FRPEFAAERTSQVKA..GHLYYWDGWNAFVINYIDSDIA..PY.EVVHLGE.I......GDKKVCERL...AAAPERI...GARVE
R5EC97.1/21-127          IKISG..KPYQAVL.FDNATG.RAIRSQL...PLK....MKMADLY.G..RELCYR..FRA.PLPTDNV..AYTRYEV..GEIVYWPPRHSFVIMYAQNGEMF....DMQKIGK.ILSPLP.............VHWSGDV...EVEIT
A0A0M2HEB8.1/20-129      FTSDR..TTVDVTIGGDNPTV.RDFLSLL...PAE....VAVEEFN.G..REKIAY..FSR.ELTTQGS.P.GSDPED..GDLIYYAPWGNIGFYYNADGIDY.SD.ATIHIGT.Y...SA..TVDQ...L...ALLEGQ.....VTIE
A5UP05.1/5-116           ITIND..NEYHADM.VDCDLV.NQIADMC...PFE....VTFKQHR.N..QEYFTK..LPS.QANDDGC.PLTTTILK..NKLYYYQQWNAFVIVYEDTNVS..PY.ELTYVGE.F.......DEDVSEYL...QEAGRNI...FVEMD
A0A0Q5H9H7.1/26-132      ISSDW..GTVDAVL.ADNEAA.SGLLDLL...PIT....LDMRDHL.R..QEKTGE..LPL.PLPRSAR...RRNFSP..GTIGLWGS.NDFVVYYREG..QVPRP.GIVVLGH.AEGDVS.........I...FDRPGRV...SIRIE
A0A139TGW7.1/53-161      LTIHH..QDFSVTL.EDNATV.TEFMKYL...PMT....ITMDELH.G..NEKYYY..MNQ.KLPTNAQ..SVDFIEA..GDLMLFGD.NCLVLFYKSFQT...SY.TYTRLGH.V.....DDVNGFIKQI...DS..NSL...QVNIY
A0A0R2BJK5.1/3-115       VKTEL..QTFDIKL.IDNAVT.AALQHQF...PLD....LTITARA.G..VEFYGK..LSD.ALATTDA.TATSQIKS..GALYYYPDWQALSFQLKDLDIS..PY.TMIYLGE.L.......PVQLVTLL..QETNRQDF...IVNLS
A0A077DDJ7.1/5-111       ITLDQ..QSFPAKL.ADNPTS.RDLYNHL...PLT....LPLDDFA....HEKIAS..LDK.RLSIEQA.PSHYQGKA..NDITYYAPWGNLAIFYGGG..PN.AK.GLIFLGR.F...QE..DVRD........VLPHAR...TIRIE
A0A0M2HEE0.1/10-118      IELPG..MHLTGSV.DHTPIA.SSLLALL...PLT....LQFTDFG.A..QEKIGR..LPA.ALRISGA.PRSSNAPA..ATIAYYQPAQSLVLYYEDV..GT.FP.GIMPVGW.LDEVTG.........L...REITSDF...TATIR
Q7VHD5.1/5-117           MQFQG..KSFVLTL.ENNAAA.RDFYALL...PLA....LSFSDYV.G..KEKIAR..LDK.SLSTQES..GEYDPQS..GDFFYFAPWGNVGIFYAKQPPYK....GLVKLGA.....PKAEKESFITHL...KAQKQDF...ILTIE
A6VKF4.1/28-139          LAKSH..EKITASL.ADNQTA.RDFYNQL...PLI....MKLEDYA.D..SEKIGR.GIPK.KLSIADS.PKGYAGNR..GDLAYYAPWGNLAVFYTDSHVGY.AN.GLVYLGK.I...TS..GLET...L...SKLDGE....KVTIK
W3AQ38.1/163-276         ISIGD..KHFTLYF.DSNSSA.EEFFEKIKKEHLV....ITMKDQG.G..VEKIGE..LPW.TLTANDE.....AVTAGPGDILLYQG.NRLSICYSET.....ST.NSTKIGH.I..PYYDDFGDFSEVL...GKGDTTV...DFTVV
A0A0X8UZ49.1/5-114       LTVDG..KRIDVEW.EDNPSV.NA.VKAFARDTLT....VPMERYG.G..FEQTGS..MER.SVVRNDT.....WTEVGPGDIVLYRG.IQICLYFGDNA.....Y.DFTRLGR.I..VGM.TESEIAEML...DR..PSV...TAVLK
A0A086YZI7.1/72-180      ITIDG..TVFQARL.ASGCAA.QEFTRRM...PTT....LRMNELN.G..TEKYHF..FDR.PIPSEPQ..AVGEIRA..GDLMLYGS.DCLTLFFKTFRT...SY.AYTRLGW.V.....EQPESLAQTL...RP..GTV...SVNFS
A0A0A0M2I1.1/75-188      INIDG..RDFSAHL.NNSSAS.RQLIAML...PYT....VRVQGLN.SGLEEHTAD..LQK.PLSTSGM.PAGAKPHP..NDIGYWSPQPRIVLYWGDV..GF.YD.GIHILGS.F...DNANAKNY....I...HSLKRPY...KITIT
C3WA77.1/37-147          VIIKN..EHYKVEL.EDNKSA.DIFLERL...PLK....IKVKELN.G..NEKYGI..ISK.KIPSDRS..YSGNIEV..GDLMLYGD.DCIVLFYKSFYT...SY.SYTKLGR.I.....VEKDRLQKNI...SETDHNL...EIIFT
A0A0G3H408.1/8-117       LIIGD..DIVGATV.WDTPTG.RDLLDRL...PVT....LTFADFG.G..QEKVAR..LDG.GLTMEGM.PSGDDPEV..GDLGYYAPNGVVVLYTGEV..GF.WN.GIARIGR.M...EG..DLSV...I...TEHHDDF...TVTIE
A0A143Z7Z4.1/7-118       LQLDG..IAYTATL.QENTVT.DMLVLQG...PIE....LLLKRYA.G..QEYYAP..LPN.PLPISGM.PTTTTVHA..GGLYCYEGFGVLSIPFQDVPVH..PY.EAMHLGD.I.......NEDILSHL...ANAGNAI...SARLE
Q6MH65.1/24-129          FTIGK..NSAIAVL.YNTQAA.REFAAQL...PLT....LTFKDFA.S..KEKIAY..PPR.KLSGKSK.....QNSE..GDFAYYAPWGNIAVFYKEE..ASATA.DLLILGQ.F...ES..GKQF.......FHVGGSF...EVKIE
B1YH45.1/41-149          ITING..QVASVQL.EDNATT.KAILAEM...PFT....IQMDDLH.Q..NEKYYY..FDK.SFPTQPQ..AIQSIEA..GDVLLYQN.NCLVIFYQAVEP...VV.PYTRIGK.I.....HYFQDIRASF...GN..DSV...SVQWY
A3XN05.1/13-124          LKVGE..KKFKAKF.NNSTTA.KDVLSKL...PYT....VNLDQYE....FDYCGI..IPA.PLAFDEA.DKHNGWTN..GDICLAD..NYFTILYAGEEQSASHT.GLIKIGE.V......EDKNQLSEI...KNLGSNI...RLTVS
A0A073IVI4.1/5-113       ITAGR..HVLKAKL.DDSAAS.RALWGKL...PLT....LPMSNLY.G..REMCFH..FGPGGLPANEA..KDQGYKV..GDLSYWPPRGSLVILYKQNGEVF....NHQTLGH.................INGDVSFFDKMDEADVTFE
A3N0R5.1/3-111           IKIGQ..QIFEAKL.ADTEAA.QQLTELL...PLT....LEMQDHL.R..NEKFAE..LPQ.NLTAYDQ..AVGSIQT..GDILLWQG.NTLVIFYERFDT...PY.RYTNIGK.I.....HNVSGLKEAL...GK..GSI...KVSFE
R5Q4L0.1/29-139          MVVDG..AIYAIEL.NASAAA.ENFAKRL...PMK....VVWEDFG.K..LERIAR..LSE.RLDVGRD.PVVKSPVR..GTFAYYVPWGNLCLFRIGG..NAPSR.DLVELGA.V.......DETALQAV....IQSGGR...EVELR
H1X9G1.1/5-117           MHLNL..ETFEFEL.ADSQVI.DELVSKK...KLK....VTMLDHY.GRLYEGDLAEAFSSDDLTTRDLVP.........GNLYYETEGKRLILPYELLKKM..DK.HYVHLAAAV......DAEAFVAFL...KQADQEI...ILSCK
A0A071M629.1/31-139      ITTGN..QVMTATF.YNNATT.RELISRF...PLT....LPMEDLY.N..REMCYR..FPE.ALPANEL..QTSGYEV..GDIVYWAPRHSFVIMYEQNGERIS...NLQKIGR.IHSGVG.........V...FRHTGNA...DVTFE
A1S956.1/5-113           LLVNG..TVIHTSL.DDSAAA.RDFFALL...PLS....LTLTDYA.G..TEKIAY..LDT.QLSTAGL.PAGTAANT..GDICYYAPWGNLAFFYRDF..GY.AR.GLIKLGR.L...PG..DCAW...L...T.DAKDI...ELVIE
A0A0J0Y5K3.1/57-165      ISSGG..RSVEAVL.TENATT.RAFITRL...PMT....LKMKDID.H..REKYAL..LSI.KLPVKS..PVEKQHVI..GDVS.YLPDGRISIIYHQDNKPT.NL.KFIKLAS.I..GATVNALSLESLS.............EISFE
A0A139TGX2.1/59-186      IQFGNEGKTFTMHF.EKNDTA.LTIARNIGEDGRN....LPIYNYDNFENYEVMQYYDIPS.SYSIPASAQTVDYQKA..GEVYYMAE.NRVVLFYQDAHI...PG.EYTKIGT.IEETEGLRSAVENNPV...QEGWGNK...LVLIR
R6Q6X9.1/56-164          IVNGQ..KTYNVSL.YENETT.KSLKNLL...PMT....VSMEELN.G..NEKYTY..LSQ.SLPTARV..YPGQIHT..GDLMLYGS.DRLVLFYEDFAT...SY.GYTPLGK.V.....DDPAGLVEAV...GS..WAV...EVTFQ
A0A0R2HL18.1/3-112       LYINN..SPQQVEW.EHNESV.EALCQRIKEKPIT....IQMRKYG.T..FEQVGT..IG..NLPADDH.....YISTLPGDIVLYNH.NQIVIFYDEH.....AY.HYTRLGRII....GKSKEELYQLL...GK..EDI...LLTIY
D5E2V1.1/57-167          LMFND..KEVLIRM.YDNPAS.MAFLAQL...PLT....IAFEDYI.G..KEKISI..LQK.RLSADDV.QAGDLSKK..GDFAYYAPWGNVAIFYKGF..EDATN.DLIILGQ.I...ES..GKEN...V...ENIHGDF...TVTIK
A0A0Q5ZRH9.1/6-114       IRFGG..ETVTATL.EPGEAA.RAFRALL...PLT....LKLTDYN.A..TEKIAD..LPR.RLPVLGE.PAGIDPEP..GDLTYYAPWGNLAIFYKDF..GS.SR.GLVRLGR.L...RR..IPDA...F...R.QPGPV...TVTIE
```

```
R5RNX6.1/26-134       IEING..QMFLATL.DDTPTS.QALLEKL...PMV....LTMKELN.G..NEKFYN..LEY.SLPVTSQ..SVNQINK..GDLMLFHD.NCLVLFYQDFLS...KY.QYTRIGQ.I.....DDAGNINQIV...GA..GDL...VVSFM
A0A162GHD2.1/66-174   LTVGD..RVVTARL.NDTPAA.RALAEQL...PLT....LAFDDLN.A..VEKTAP..LAS.PPSMAGM.PSGDDPEV..GDIGLWAPSGDLVLYYGDV..GY.WD.GIARLGT.F...D...DVEA...I...ASLTGPF...TGTLA
C6LE92.1/245-354      LVLDE..GEMIVEL.YENSAS.DDLLERL...PMT....IGFEDYN.G..TEKISY..LDS.ELDLSNA.PGECTPQA..GDLTYYAPWGNLAFFYQDF..RN.SP.QLIPLGR.I...ET..GGEY...L...ENLDSYT...EVTIE
B2J892.1/63-173       IKVKD..KVVTAAL.IDSKTT.QDFVSLL...PLT....LTMNDLF.G..REKFAH..LPR.AISEEGE..RTKTYEV..GEVIYWSPGPDVAIYYRHGGEEIPDP.GIIVIGK.I...DS..DLEA.......FNLPGSL...KVTIE
R7CV31.1/238-346      IIVGE..QTITATM.EDNGAA.RDFLSRL...PLE....VTLEDYN.NG.TEKIFY..PDP.ELSLDDT.PRGCTPAV..GDITIYEPWGNVAIFCRDW..SE.SS.SLIEIGH.I.......DDDG...I...SLLQGTE...SVNVR
R2PZ68.1/5-113        MFIGD..HPYPVLL.NDSQAA.KDFYRLL...PLN....VTLNEYN.G..TEKICH..LSE.RLDVRDS.PKGMTAES..GQLNYYTPMGNLCLFYKAF..PY.SQ.GLVNLGS.M...EE......VIPF...EKYSENI...FVSFR
F5XMK7.1/66-175       IVVAG..KTLNATL.SDNPAA.KSLVAQL...PLT....LDFADFG.G..QEVTAE..PPR.PLTMEGM.PDGESAPA..GTIGYYAPDGVVVLYYTDV..GR.YN.GIVRLGR.I...DG..DISI...L...KGWDEAR...PVTIE
D7JDK2.1/61-168       LTVGN..KTFTATL.DVNASA.DAFRAQL...PLT....LDMTDYG.G..FEKIYK..LDV.KLPYNDK....LEESLGLGDIMLYQS.NTIVLFYDNHG....GF.SYSRIGK.I.....DNTLGLREAL...GT..GNV...TVKWE
A0A0M3C492.1/46-155   ITSGD..KVVKAIL.YENPTT.KDFIAQL...PLT....VDMEDFA.G..KEKIFY..PPK.KLSTAAR.KAVSDPKI..GDINVYAPGGNIAIFYGSY..SG.SR.DLIRIGR.I...TE..CIDV...I...NVTGTVK...NVYFE
X5E891.1/8-117        ITIDG..NEYAAEL.QDNPCA.VALKNEL...PVT....LTFQDFG.G..QEVLAE..APK.KLPMSGM.PASAGASP..GDIGYYSPTGSIVFYYASV..SP.FP.GIARLGH.F...TE..DVSF...L...ASTRENL...EVTIS
I4N7N7.1/60-168       MRAPQ..GTIEFKL.DDNKSS.RDFASML...PLE....ATLEDYA.S..TEKISY..LPR.KLIIQDA.PDGYTPRE..GDMAYYAPWGNFAIFHKDF..TY.SI.GLVKLGT.L...LS..GMDI.......LRKKGPV...QVKME
R7KJS8.1/26-135       MIIGE..QAFTIEL.LEHDAA.RAFADRL...PMT....LTFENFG.S..TERIAY..LKQ.SLTIGSA.PTSADPKV..GDLAYYIPWGNICVFVKDF..RH.SE.DLVPMGK.M.......SFEATKAL....KESGNQ...PVTFR
A0A0Q5B8K5.1/13-121   LEIEG..VEVTAHL.DGSATS.ASLLAWL...PLT....LPFRDMG.G..QEKLAD..LGD.ALSLDGA.PSRSDARS..RTIGYYSPARSLVLYYEYV..GA.FA.GIVPLGS.FDDVEP.........V...RDLADGT...HVTLR
R2SMP4.1/59-171       INVND..AEFTIQM.NDTATG.RALVSMI...PSTSMRLPTSYEQE.G..VLKYYD..MAR.EVVS.DP.EELSSVSA..GEFLLDGN.DRLLLYYEDTELN..G..SYTRVGR.I.....EDATGLAEAL...GD..GDV...VFTVS
C0QZ61.1/28-139       LTFGS..NEIYALI.TNSKAG.NDFLSLL...PLN....IKAEDYN.S..TEKIFY..LSK.KLNTQNE.PDGINPKA..GDITYYAPWGNIAIFYKNF..RY.SN.NLIYLGK.F...ENASDISK...L...SNMKGDF...DIRIE
C6BTC7.1/44-155       LIVGE..TVIPALL.NDSKSA.QALIAKL...PYT....VELQRYA....HDYCGV..MSD.GLPYDKS.DLRDGWLD..GDIAFAVSGNYFTILYKDEDISEQFD.GIVNMGI.I........KAPLSIM...DTLAESI...SLRIE
Q8EWN5.1/54-164       INING..QPFNTIL.ESNSTV.NSFLNLL...PLS...NLNMNDLN.S..NEKYIY..LSE.TLNTNTY..KPGIINA..GDVMLYGN.NCLVIFYKTFTS...NY.SYSKIGT.I.....ENVDELVDLL...DTR.NSV...NVSID
A0A0S2W3S0.1/1-108    M..DD..RTCAVTL.YDTPAA.ERLYEML...PLE....LTFEDFN.G..TEKIGY..LPQ.PLDTGEG.ANGVDPAV..GDLCLYAPWGNLCIFYQDS..GY.SD.GLLPLGR.I...ET..GMDL...I...TEMDSPF...TATLE
A0A0D0SGX5.1/68-178   LDFDG..EQIEGVL.DNSKTS.ENFLKLL...PLT....LDMTRFY.D..REYAAG..LGE.TLSQEGK..IIDDFEN..GDITYYIEGNALAIFFDKADSSD.QG.GLIRMGK.I...TS..DLDK...L...IQMDGDR...KVTIS
K8E2X1.2/3-111        IIIGK..QLFQIEL.VSNQAT.KELIARL...PIN....LKMNDLH.G..NEKYAY..FSE.ILPTQEE..KVDEIKK..GDIMLYGS.DCLVLFYKTFST...NY.SYTKIGK.V...KEVDQLDFISEI.......ETI...NVILV
R5PZ69.1/30-141       ISFEG..TEVIIKP.EDNSAV.KEIIKML...PAT....LEFSDFA.G..EEKIAV..LPE.PISLDGA.PRGMKASA..GKVFIYAPWGNFGIFYKEHGRSI.DQ.SLIPLGE.V...EK..GLES...L...ALKRGGF...KAKVE
F2NTA4.1/35-141       VSDGK..NKIVYEL.NASGQS.KSLYSQL...PIK....VQIENYS.T..NEKIFY..PKE.KIPLKNG.IEG.SGDS..GTLAYFSPWGNIVLFYGKF..SG.YP.GLFILGK.A...VS..GAEN...I...KNLSG.....IVSVE
R9M0V0.1/46-155       LTAGE..TVLSGVL.FDNETA.RAFAELL...PLD....APLWDPAPG..YARAFD..LPR.RITDAPV..RTRAYEL..GSLAYWDEGPSIAAIIYNDNREET.VV.PVTAIGR.L.......DGDVSIFFGYDQ.........PVHIE
A0A0D6XR27.1/6-114    LTIHD..QTYTATL.NENPAT.EKLMRLL...PLK....IKMTDLN.R..NEKYHT..LDT.TFPTQQE..AVQQIHS..GDLMLYDN.DTIVLFYQDFST...PY.TYTRLGK.L.....IDATSLSDHL...GY..NDV...TVTIA
A0A099VS82.1/27-135   IEFDN..QKVLIEL.EENATS.KAFVEML...PLE....LEWSDFA.N..KEKITY..LPS.KLQAKGD..SSYIPQI..GDFFCYAPWGNVGIFYEKQ..PP.NS.GLVFMGK.V...........KNGLGILKSQNKPF...KTQVY
A0A0R1XD82.1/68-184   IQIGK..QIFRAHL.NNSVTA.KAVLAKL...PVT....LTVQGLT.TNPNEHTAA..LKR.ALPIKGT.PTGADPAP..GDIGYWAPEPSLILYWGDV..DY.FN.GIHILGR.FDQSDRQTAIRY...I...HQQQAPY...QVIIS
A0A0H4R3U6.1/16-128   IQVGK..KHVKGTL.NNSSAA.KSLQKKL...PLS....LAVKDFPGE..PEKNAD..LNF.KLSTDGM.PKGSAAKK..GSIGYWSPDRRLVFYYGKV..SY.YQ.GIHIIGH.F...NSKKDLKT...V...KNIKKNQ...KVVIT
R6ZSB6.1/6-116        IQIHD..HDLLVEM.IDNSSS.KALIKRLKQSDIV....LEMKEFA.N..MEKFGV..LDK.KYPQNDE.....WVTTLCRDVILSEG.YLLVIYYAPN.....TW.NFTKIGKVI....NVSDEEFKRIL...GK..GNV...HARIH
C0EAD8.1/27-137       MITEN..TQVVITL.NGSRAA.ADLAAML...PLE....MTLIERN.S..FAKGMT..LPE.HLSSAEA..TTREYEI..GDFGYWSAGPDLAIFYDDIYEQT.IV.DVIPLGH.........AETGAETM...ANERGTV...RLEFV
E0DET7.1/58-168       VTING..QVLHARL.WDNAPA.RDLLDRL...PLT....VRLQDVD.N..QEKVGY..LPKPPLSADGM.PEGDDPQP..GDIGWFRPWNTLAFYYGDV..SY.SG.GIARIGR.F...DD..PIDL...V...KAQTGFF...HATIE
C9A5L3.1/54-164       ITVGQ..QDFIAKF.YENEAS.EYLMNQM...PFT....LTMSDLN.N..NEKYYR..FSE.NLPEMTT.ERPEIIHE..GEIMSWNS.HTLVLFYQTFTN...SYGGYSRIGV.I.....EDPAGLREAM...GE..EDV...EVVFS
E9SFQ6.1/174-285      FDVGD..RTFLAKL.EDNSSV.DDLISKMPVSGIE....ITMSDYG.G..FEKVGD..LPF.ELTTNDT.....DITTVPGDVILYQG.NKITVYYGEN.....TW.NFTKLGH.I....DASREELLEAF...GDGETEV...RISVE
A0A0M0G6F7.1/52-162   AEIGG..EEVEITM.YDNPTS.SDFIDQL...PLE....LTFKDFG.G..FEKLSY..PPK.KLTTEGA.PEGDTPSA..GDFAYYAPWGDVTLYYKDE..SY.AK.GVVLMGR.M..EDG..GIEK...V...AGMGEDE...VVRLR
E8LJ20.1/19-129       MTIND..TKYVVTV.DENTAAGKLFLEVL...PLS....LNFENFG.S..NERIAY..LPH.KLDMNSY.EEPISVKR..GGMTYYVPWGNLAVFRKSF..SC.SA.DLAPLGA.M.......SEEAISAL....EKSGSA...DVSFK
A0A0S9RD28.1/60-169   IEFAG..DQVEATV.LDTPVG.RDLVAQL...PLE....LDMSDHG.G..VEKTGP..LPR.ELSTDGE.PTGADPDV..GALGYYAPYGDLVLYYGDQ..SY.FD.GIVVLGR.M...GK..GFDA...L...GRIDGNV...SVRVE
A0A0D5LWJ1.1/5-113    VIVGD..TTLSASL.DNSPAA.RDFASML...PLE....LTLSEYA.G..NEMVAD..LGR.KLDTTGA.PASYKPKT..GDITQYSPWSNLAIFTKPF..SA.SR.GLIRLGE.F...DG..PIDA...L...T.VGGNV...TARLE
A0A0Q5QLJ1.1/59-167   IRFGG..TMLTGSL.DTSPAA.RALRDRL...PIT....VAATDYG.G..VEKTAE..IPA..LPMVDM.PAGADPEP..GTLGYYAPDRVLVFYYGDV..GY.FP.GIAALGR.F...TD..TDGV...V...ATATGAV...TVTVE
B1Y0L5.1/59-168       MTIGGT.HRFAVTL.ENNPTA.RAFAQML...PLT....LDMPDLN.D..NEKHVR..LPH.SLPTHAQ..RPGTIRT..GDVMLYGS.DTLVVFYKTFPS...SY.SYTRIGR.V.....TPVDGLVQAL...GT..GSQ...RIGFA
A4BGH4.1/36-145       LQFDG..QEIAIEL.DESATV.DSLLAAL...PLT....LTFEDYA.G..KEKIAH..PDI.QWDTADA.PAGYDPSV..GDLTVFAPWGNLALFYGEQ..SY.AR.GLVYLGK.I...VR..GADQ...V...STLDQVA...QVTLV
A0A160IS58.1/11-119   ITIGD..EEFSTRL.YDNQTI.RALIEKL...PLS....IVMEDLH.R..NEKFYY..FSE.KLLTESV..IPGNIKA..GDIMLYGD.NCLVIFYESISS...SF.SYTRLGY.I.....DDVEKFAQAV...GD..GDI...HVSFD
R7I7N8.1/7-115        FTAGG..RTFKVEL.ADTPAA.RAFRKVL...PAK....LPMLELN.G..NEKYFH..FRDRTFPAAPA......VHAKAGDVMLYQD.DYVVIFYMTPENS..PY.TYTRIGR.V.....TDTKDLIRAL...GS..GNV...DVSWE
X2GXM6.1/20-128       LTIDD..QQVIVNL.LGTPAS.HQLLALL...PLT....LTFSDYV.G..AEKIAY..LPQ.RLITQGM....ASAAHISGDFTYYAPWGNLALFYQGV..GT.NS.QLYTLGH.I...ES..GRSV...L...ANLKQDF...VATIS
C7N7E5.1/6-118        LVFGN..TEVFAQP.NDSQTA.KAFAEKL...PVT....IPVGGTG....IDFCGR..MPF.ALPYDEA.DVHSGWVN..GDVNYNPHGGWFAVLYGDEEHSGRYG.DQVVMGR.I.......EGSELAKV...QSLDGDF...DLRIE
R6UDI3.1/50-155       DE.HD..TTVRFVL.FSSPAA.KSFYEQL...PLR....VNVEDYS.D..NEKIFH...PLKELETQDTPQAVASS....GTLAYYKPWNNIVLFYENGQP...SD.TLYALGEAVSGKDNIKRLQG.................MINIT
F4GK14.1/134-243      ITVGS..TTFTATM.ENNASV.KALMELLVKEPLT....IQMSEYG.G..FEQVGS..LGQ.RLPSNDL.....QTTASAGDIVLYSS.NNIVIFYGSN.....SW.SYTRLGK.I...ESAGAKEIKDAF...GSG...A...SVTLS
A0A172Q5T8.1/59-166   ARIDD..QTFEIVL.NDSQAA.EEFKELL...PLT....VEMEHVN.G..NEVYAP..LGE.QFTANNQ..QAGQIHA..GDLKLWSG.DGLVLFYKDFSS...NY.SYTDLGR.M.....TDSKGLADAL...EQ...SS...TVRFE
G9PSJ7.1/32-140       LTAGG..KSFTAEL.NDGPAA.ATLLKKM...PFT....LEMKDLN.E..NEKFAY..LDG.ALPAKAS..APGSIRA..GDIMLFGA.DCLVVFYKTFPT...AY.SYTPLGK.I.....EGAEAVSALS...GK..DGI...KITFS
A0A0F7D3K8.1/26-134   MNIDD..QQFEVIL.HDNPAA.KAFVNTL...PLQ....LGMEELN.A..NEIFAD..LPH.KLPSSPV..RPGTIHA..GDLMLYGT.QTLVLFYASFES...SY.RYTPIGK.V.....IHPENLPAMV...DK..KKI...GVRFN
D2RJS5.1/9-117        LLAGS..KVYALKW.ADTKAA.AELRQQL...PLA....KTFTELH.G..NEKYYK..LPQ.HLTAADE..DVREIHK..GDVMLFDG.QYVVVFYQDFQT...TY.RYTRLGR.V.....EAANDLDAAL...GA..GDV...FLTLQ
A0A091C5L1.1/47-156   MTIND..ESYPVTL.NDSEAA.QNFVDMM...PLT....LTLSDYA.D..TEKVSD..LPS.ELELGNS.DRGHQPSP..GDITIYEPWGNLAIFYNAF..DY.SD.DLIHLGH.I...ED..GADM...L...ESNEEEF...DVTFT
R6N163.1/233-345      LTANG..KSFTATL.VENSST.EALKARLSQSNIS....IQMNDYG.D..MEKVGS..LGF.SLPRNDQ.....QTTTDPGDLILYQG.NSFVIYYDTN.....SW.NFTRLGK.I..DGVSTREQVLDLL...GGK.GEV...TVTLS
```

```
R5B743.1/25–135         IIANG..QTMTATL.ADTEAA.RQLLTRLDNGPVT....IRMNDYG.G..FEKVGS..LPW.SLPASNR.....QITTTAGDIMLYQG.DNIVIFYGSN.....SW.SYTPLGR.I...DGAGVSEIRDFL.....SGNSI...NVTFA
R6UDI3.1/178–285        L.IGT..KTFTLSL.HDSESA.RAFLKRF...PMT....VTIQELN.G..NELFAY..MDE.NLPTDAQ..RATKIHT..GDVKMFGR.DCPMLFYKDFAT...AY.SYTSLGK.V.....DDAEALAQAW...KE..GNK...EVIFT
A0A136Q1G1.1/240–349    FSFDG..GEAAAVL.NDTPTV.QSLLAQL...PAT....VTMSDYA.G..AEKIAY..FAE.ALTDEGA.PEGYDPQI..GDVACYGPWGNMAVFYNDQ..PY.AE.GLCPMGK.I...ES..GMDL...L...AALPEDA...SVTVE
C5T138.1/53–161         TTADG..VKAQATL.LDNATA.RTFAAKL...PLK....VRMGDHF.G..RELYGP..MPTIAVSD....PLRKTYQA..GDIAYWPPAPGFAIYYTVGGPVIPGD.GLALLGS.IDTNLD.............IFSRGST...EVTIE
V7HXV7.1/58–170         VSFNG..RSFTATL.NDSPVA.RAIQKQI...PFT....VSFIAYGNG..QEKIGDLPFTP.QLGNYNY...DDNGQK..GKLAYWQPDNRLVLYHGPVGS...YP.GIKVIGS.F......DNAKAVYAL...KKMADNT...EVTFS
A0A154BNQ1.1/66–184     ITAGN..TSMYATM.KDNRTA.QDFIELL...PLK....LKAFDRI.G..LVKSTV..LPH.SISDDGE..RTRKYAI..GSIFYWPEGPEVAFCYSDHLPKT.VV.DIIHIGM.LESDVEHFRNYTGELV...VELANEV...PVQVE
R5A731.1/84–192         IDVGG..QVFYGEL.RDTEAA.KALKEML...PMT....LEMTDQE.G..MSKRFE..LPS.VLTQTEE..EYPSVQE..GEVLLEGS.GTLCFFYQEDSQGG....TYTPIAT.V.....REPEGLSQAL.....AGERV...EVSFQ
A0A0R2FTF4.1/39–152      VRFSN..HDYKAHL.DNNAAA.NGLKKKL...PFK....LKFSAFG.SGFDEKIGD..LPA.KLSTKGM.PNGNSAQT..GDIGYWSPQPRVVLYDGHV..NY.YA.GIHIIGH.F...DSKKAVQA...L...KNSRVHL...QLKLG
T0RXV5.1/5–113           LSQQG..IVIEIEL.EDNPTS.RELFDQL...PLK....VDIEDYA.S..NEKIFY..PPK.KLSTAGA.PVGYEPCE..GDITYYSPWGNVAIFYKDF..SF.SN.GLIKMGR.I...SS..GLDH...L...KSLNYS....EVLIE
A0A089KS45.1/6–115      IKMAN..INFTAIL.YDNESA.RTIVQEM...PFT....LNMEDFA.L..QEKIAE..LTF.PLPSAQT.ETPATIKA..GDLSLWSG.NNLVLFYTTFSN...AY.RYVPVGY.I.....EDVTGLQSAL...GN..GTV...TLTFS
```

B.1.2. PF18701

```
X1X051.1/386–481          KRWQLVQGMTQGFWRRWSSEYLRSLQPRTRWTTADKLSIKI.GDLVLVIEDNQPPLKWHLGRVMKLHPG.L.DSI..VRVVTIQMSGGRM.............FQRPIVKLCC.PL
J9LX59.2/414–508          KWWKLVQKAFQLFWRRWREEYLSSLQGKSKWTQSPSQ.IEI.GTLAVLKEDNSPPLSWRLVRVTATHPR.Q.DGV..VKVVTLRTPSGTE.............ITRTAVKIC.PL
X1XU04.1/252–345          TRWQLLRQLHQSFWKRWAQEYLNTLQGRQKWTAIQDS.LKV.DDLVIVEAPSQPPSVWRMGRITAVHPG.P.DET..VRVVTIKTQDG.E.............IKRPVVKVV.KL
J9L9H5.2/1935–2028        NRWQLIRQCHQSYWKRWSREYLSTLQGRQKWFKASPN.LAI.GDMVIVEAPSRPPTEWRLGRVLEVHPG.S.DDV..VRVVSVRTQDG.V.............YKRPVVKLV.RL
J9LPL8.2/1121–1214        GRWQLLRQAQQSFWRRWSHEYLHTLQGRQKWFRQTPN.LMV.GDLVVINTPSRPPMSWQIGRIIEVHPG.E.DNI..VRVATVKTQEG.T.............LKRPVIKLV.KL
J9K351.2/1776–1869        RRWQLLTLFHQSFWTRWASEYLTSLQNRAKWIRPQLN.IEV.GDLVIVRCPNLPPTAWKLGRVESTHPG.D.DGV..VRVVTVRTTDG.T.............FKRPCVKLV.VL
J9JUK4.2/1605–1698        NRWELLRQIYQSFWKRWASEYLTTLQGRSKWVQHQPN.VKV.GDLVLIQTPNQPPMFWKLGRIESTHPG.Q.DGV..VRVATVRTNNG.S.............IKRPVVKLA.VL
X1X8C2.1/3255–3348        TRWKLLQQSFQFFWRRWSREYLNTLQARGRWTKADTN.LEV.GTMVIVKVNDAPPLSWPLGRIIEVYPG.T.DKV..VRVAKVITKQG.V.............FTRPVVKLV.PL
J9LZV0.2/980–1073         NRWKLLHQVFQAFWRRWSNEYLHTLQTKGRWVVNQEN.IKL.GELVIIKDNTSSPLLWKLGRVQELLPG.P.DRV..VRVVKLLTKQG.L.............IIRPVVKLV.PL
X1X3D9.1/1675–1768        QRWKLLDQCHQAFWRRWSTEYLTSLQGRSKWTTEAPN.VKV.NDMVVVIDNQSPPLAWRLGRILEVLPG.N.DGV..VRVVRLLTSHG.Q.............ITRPVAKVV.VL
J9L3C8.2/815–908          DRWKLMDQCHRVFWRRWSSEYLTTLQSRPKWTEWVPN.LSI.NDMVVVIDSQSPPLLWRLSRVTELLPG.S.DGH..VRVARVLTRVG.V.............VTRPVVKLV.KL
X1XAP5.1/210–299          NRWKLL....QSFWRRWSSEYLCSLQARTKWTNNVPN.LKD.GDMVVIKDNQSPPTAWRLGRVLNVMPG.A.DGV..VRVARVLTAQG.E.............FTRPVVKLV.LL
J9JKU0.2/1628–1721        HRWKLLHQCHQSFWRRWSNEYLCSLQTRNKWTSQGTN.LNV.GDMVVVKDHKGPPTSWLLGRITSLAPG.K.DGV..VRVVKVLTSQG.E.............FTRPTVKLV.LL
A0A0J7KDN6.1/1585–1678    DRWQLLQRMFQDFWKRWSLEYLQTLQQRPKWHSKGVN.LSD.GHLVLIKETNAPPLRWKLGRIVELHQG.Q.DAV..VRVATVRTAGG.T.............LTRPLVKLC.PL
A0A0J7JVD5.1/102–195      DRWQLLQRMLQDFWQRWRMEYLQTLQQRPKWLKQVTP.LAE.GALVLVQEANAPPLQWKRGRITQLHPG.R.DGI..SRVATVRTAEG.I.............LTRPLVKLC.PL
A0A0J7KD57.1/932–1025     DRWQFVQQMHQVFWKRWHVEYLHTLQQRPKWLKPLDP.IQE.GALVLIKDEHASPLRWKRARVEGLHPG.T.DGI..SRVATVRTADG.V.............FTRPLVKLC.PL
J9KD60.2/1783–1876        KRWSLIQQIQYHFWTRWKNEYLHTLQERPKWNRPDKN.LQL.DDLVIIKEP.TPPLKWSTARVIEVHPG.D.DGI..VRVAKVKTSTEKV.............LTRPAVKLC.PM
X1X039.1/897–989          QRWRLVTDLHRHFWSRWKNEYLSSLQARSKWFGNVDQ.LRE.GALVLIKEA.SAPLHWRLGRIRTVHPG.S.DGV..TRVATVDTSTG.S.............LTRPAVKLC.PL
J9LKZ0.2/640–732          QRWRLLRDIHQHFWSQWKSDFINTLQQRVKWTKDNGA.LRQ.DDLVLIKEP.ISSLQWRLGRIVQLHPG.N.DGI..NRVATVKTTSG.T.............FKRPAIKLC.PL
X1WYQ8.1/1613–1705        QRWRLIKDLHTHFWKRWQRDYLQTLQRRSKWSLGQEN.ITV.DTLVLIREP.TTSLSWKLGRITQLHPG.L.DGV..VRVATVQTANG.L.............LKRPTVKLC.PL
A0A0J7ND98.1/1176–1269    SRWQLIRQLTEKFWKIWQADYVNALQQRSKWRHQQPP.INI.GTMVLLRNPSLPPCKWELGRVTQTFPG.S.DGL..VRVVSVRTAQS.E.............YQRPIVKLS.VL
A0A026WVN7.1/402–495      SRWQRVRHLTERFWKLWATDYVNTLQQRQKWRKEQPS.IKP.GQLVLVRNALLPPCKWELGRVTQCHPG.A.DGC..VRVVSVKTANL.E.............LKRPIVKLC.VL
A0A026WMA2.1/239–332      SRWQLVRLMTERFWRLWQDEYINSLQQRAKWRTVGPL.IQV.GQLVLLRSPLLPPCKWELGRIIQTHPG.S.NGL..VRVVTIKTATS.E.............YRRPIRKIC.IL
A0A026W1L0.1/83–176       SRWQIVRQLSERFWKLWQTDYLNTLQQRAKWREPTSS.VKV.AQLVLIRNDPLPPCKWELGRVSQCHAG.D.DGF..VRVVTIRTTTS.E.............YRRPIRKLC.VL
A0A0J7MQW8.1/239–332      SRWQAIQRMYERFWKVWSSDYLNALLQRKKWRSSQPN.VQI.GDLVLLRHPNLPPTKWELGRVVQCHPG.D.DNL..VRVVTIRTAKS.V.............LKRPITQLC.KL
A0A0J7N681.1/140–233      SRWQTIQRLQERFWKIWSADYLNSLQQRTKWRTKQEN.LQV.GDLVLLRHSNLPPTKWNMGRVIECHLG.V.DQL..VRVVTVRTAKS.I.............FKRPVTQLC.KL
A0A0J7KDW6.1/219–312      SRWQQVQSMHERIWRSWSHDYLHSLQQRRKWTESQPD.VKV.NELVLLKNNLLPPSKWGLARITEVHPG.P.DGR..VRVVTVRTAES.T.............FKRPIAQIC.RL
A0A0J7JXM4.1/141–234      SRWQLLQQMRDQFWIRWHQEYFQGLLTRTKWRQDTTA.FRP.GQLCLLLHENTSPTRWPLARITAVHPG.N.DGL..IRVVTIKTSSS.E.............FKRPVTKFV.LL
A0A0J7K420.1/435–528      TRWQLLQKMRDHFWEKWSREYINGLTSKSEWLKANAA.PDV.GALCLIKSETIPPSRWPLARIIQLHPG.D.DGI..VRVVTVWTPSS.E.............LVRPLTKVV.LL
A0A0J7KK97.1/169–260      SR..HIQRMRDHFWQRWSQEYILALTPRPKWQKREKS.PDV.EALCLVRSELTPPSRWPLARITKLHPG.N.DGV..VRVVTIRTSTS.E.............LVRPLVKLV.LL
E2AMZ8.1/108–201          SRWQLLQQMRDHFWQRWFQEYIHSMTSRPKWLKDNRP.PKV.SVLCLIRSDITPPTRWPLARILKTHPG.E.DSV..TRVVTVRTSSS.E.............LIRPLTKIV.LL
A0A0J7KC49.1/220–313      SRWRLLQQMRDHLWQRWSQEYLQALAPRPKWWSTTGQ.LRE.GQLCLVKNENTPPCRWPLARIIRLRPG.D.DGQ..VRVVDLQTGQG.E.............LTRPVVKLV.PL
A0A0J7K9I1.1/154–247      SRWQLLQQMRDHLWQRWSREYVQGLAPRPKWWATPGN.LKD.GQLCLVKTETTPPCRWPLARIIRLHPG.E.DGH..IRVVDVRTSGG.E.............LTRPVVKIV.PL
A0A0J7MVI0.1/220–313      TRWQLVQRMRQHFWDRWSRDYLHSLNHRPKWWKSDAA.VRV.GRLCVLRNKTTPPNKWPQARVVDVHPG.E.DGH..VRVVTVRTASS.T.............FRRAVNRLI.LL
X1WTY8.1/895–989          NRWKLLQQQQQQFWRQWSSDYLHTLQQRLKWRTPRCN.LQV.DDLVAVKDDHTSPLQWPLARVIALHPN.THDDL..VRVVTIRTAHS.T.............YKRPITKLI.KL
J9JW47.2/1200–1294        TRWNLIQQLQQEFWKRWTSEYLCSLQNRPKWRFTQPN.LQV.GDLVIVKGIQTSPTQWPLARITQLLPSKS.DGQ..VRVVKLRTAAA.E.............LTRPITKLI.KL
```

```
A0A026VVE6.1/82-175       SRWQLLRQMIDRFWRRWSTEYLQRLQARNKWQHPERQ.LTK.GSLVLIQDERFPPSKWPLARVIDTHPG.T.DGL..IRVVTMRTTVA.T.............YKRPVHKLC.PL
A0A0J7KB34.1/1636-1729    SRWQLLRQMLERFWARWSTEYLQRLQTRNKWHLSTKP.LQI.GDLVLVLDERYPPAKWPLARVTALHPG.T.DGR..VRVVTVRTAVS.E.............YKRPIVKLC.PL
A0A026WPM6.1/82-175       SRWQLLQKVRDQFWRRWSSEYLQRLQDLSKWQRPVKS.LGV.GSLVLLADERYPPSRWPLARVTEVHPG.Q.DGL..VRVATVKTQSS.T.............LKRPIVKLC.PL
A0A0J7N900.1/910-1003     TRWQLLRQAVEHFWTRWSSECLQRYQAISKWHHPANE.IKE.GSVVLITDERYPPGKWPLARVIQLHPG.P.DGL..TRVVTLRISTS.I.............YKRPIAKLC.VS
E2AKS2.1/97-190           SRWQLLKQMLDTFWSRWSQECLQRFHDVSKWNKPVPS.LKK.DSLVLVVDERYPPAKWPLGRVIDVHPG.A.DGH..VRVVTVRTQTS.V.............LKRPIVKLC.PL
A0A0J7KCX3.1/1242-1335    SRWQLLRKMTESFWTRWSTEYLHQLQVSNKWFKTQPI.LKL.GTLVLVKDERLPPSKWALARVIDVHPG.A.DGL..IRVATVRTQTS.T.............LKRPLVKLC.VL
A0A0J7KIZ4.1/613-706      LRWQLITSMRDHFWTRWSKEYYQHLQQLGKWRDRAVN.LEI.GTLVLLKDELLPPAKWALGRIREVHPG.S.DGL..VRVVTVETASS.R.............LTRPVTKIC.PL
A0A0J7K9N2.1/1021-1114    SRWALTSAMRDHFWKRWSSGYVHHLQQLRKWPKRAPN.VTV.GGLVLLKNELQPPTRWALARIIALHPG.T.DGL..TRVVSVRTANA.T.............FKRPITKIE.PQ
A0A0J7KD18.1/636-729      TRWRQVSQMRDHFWQRWSREYLAHLQQLPNWRQQRRN.LDV.GDLVLIRDDLMPPAKWSMGRVAEVHPG.P.DGL..VRVVTLRTARG.S.............TKRAIAKIC.PL
K7JL55.1/231-324          SRLQLLQRIRNQFWNRWSSEYLLHLRQREKWRNPTEN.FYI.GRLVLIKDDRYPPSKWLLGRIVEVHPG.P.DNL..VPVVTAKTATI.S.............LRRHVARLC.PL
A0A0N0PAG9.1/1422-1515    KRWQLTQRMLQDFWRRWSLEYLSQLQNRYKWAHCVPE.PKI.GDVVLVKEDGLPPSKWLFGVIVDKHPG.L.DNL..TRVVTLRYKNN.E.............IKRPVSKLI.VL
A0A0N1IHU1.1/1606-1699    DIWTLVQKMTQHFWKRWSTEYLQGLQRRQKWQKSSIS.PDV.GQLVIIKENDLPPAKWLLARILELIPG.P.DKA..VRVVKLKCKSS.I.............LTRPINKII.LL
A0A0L7QJ67.1/161-255      NRYQLVQKTIQHFWKRWQREYLHGLQQRHKWRTDSPDLIRI.GAMVLLREDNLPPLQWRLGRIIDLHPG.P.DGV..TRVVSVRTAHG.T.............LKRPVKKIC.VL
A0A0L7QYD7.1/223-317      GRYQLIQQMLQHFWRRWQREYLHQLQQRHKWTQSTSPKVTE.GALVIVKEDNLPPLLWSLGRFIHLHPG.K.DGV..ARVATIKTAEA.E.............YKKPVTKLC.QL
A0A0L7QKY5.1/94-188       KRYQLLKQMFQHFWSRWSMEYLHNLQQRVKWRRSTTLGFKA.GDMVVIKEPNLPPLKWMLARITETHPR.P.DAV..SRVLTLRTAEG.I.............LKRPITKVC.FL
A0A194RI04.1/1633-1726    ERYRRIELLKQHFWRRFCNEYVTLLQQKLKWHSSKSE.LKL.GSLVLVKEKALPPLLWCLGRVIQLYPG.S.DGT..ARVAELKTRRG.T.............IRRAFNNIC.PL
A0A0N1IBH6.1/1626-1719    DRYKRVQKLKEHFWDRFSLDYVHLLQQRVKWRAPSTN.LPL.DTLVLVRERNQPPLLWCLGRVVGVHPG.R.DGV..TRVADIKTRKG.V.............IRRAFNNIC.PL
A0A194RAG7.1/1635-1728    DRYRWLETMRQHFWERWRNEYLSELQQKTKWRVKQRG.LQE.GELVVIKEANVPPLKWRMARVHKLYPG.S.DGI..ARVADVYTAKG.I.............IRRGVHNLC.SL
A0A194RGQ8.1/1603-1696    NRYEAIEKMRQHFWDRWRKEFIAELQERTKWKTRQQD.LAV.GDMVVLKEDNLPPLLWRLGRVLQLFTG.P.DGV..NRVADIKTSRG.T.............IRRAFNKIC.LL
A0A194QJ48.1/1595-1688    SRFQHLEQIRQHFWKRWTNEYIAELQQRSKWRAESRH.LKI.DDLVLVKEDNAPPLCWRLGRVNRLYPG.P.DDV..PRVADVVTSQG.T.............VRRALNRLC.LL
B4IRI8.1/142-235          DGWQRVSYLQQLFWSRWKEEYITSLQQRSKWRTAKPS.LAV.ADLVLVKDENLPPMKWPLSRVVELLPG.R.DGV..SRVAVLKTSSG.I.............TKRAVNKLC.LL
A0A0T6AWJ6.1/217-311      SNYHHVQLLAQHFWKRWSGDYLHTLQQRSKWQLGSKS.ISLEGALVLLKEDNLPPMLWKMGRITRLFPG.N.DSV..VRVVEIKTSGG.I.............FKRAVNKVC.IL
A0A0J7KFV3.1/219-313      TRYQLLIQIKQRFWTRWSKEYVAQLQQRVKWKLSSKTRIKE.GTMVLLKNENTPPMSWPLGRIISLHPG.A.DDI..TRVVTIRTSRG.I.............VKRALSKIC.IL
D6X1G6.1/1380-1474        SRWQLVEKIRQTFWKRWRSEYLSQFQVRNKWKTASTISLQP.GTLVILREDNLPPLHWQLGRITETHEG.A.DGI..TRIVSVKTASG.I.............TKRGVNRVC.VL
D6WI41.1/218-312          NRFEYQEKICQDFWQRWHQEYLSYLQQRKKWTQSTRQ.IRP.GDLVVIRDQNLPPMRWKMGRVEEVYPSPS.DGV..VRVASVRC.AGKI.............VKRACNRLC.VL
D7GYB7.1/1488-1582        SRFQRLQRIVQHFWSRWTREYICELQNRSKWRNNSLNALKK.GSLVIVRDDQTSPQQWRLGRVLDLHPG.H.DGI..VRVVTIKFATQ.I.............AKRPVAKLC.VL
A0A085MU62.1/717-810      RKWRHAQAIVEGFWKRSLKEYVPCLIERRKWLRPARN.LSV.GDIVIVVDPQCPRGHWPLGRVIECKPG.H.DGV..VRVAKVKTKHG.I.............YVRPVTKLC.LI
A0A085MU05.1/301-394      RKWRCAPAIVEAFWKRWLREYVPCFIERRKWLRPRHN.LEV.GDIVLIVSPQSPRGHWPHGIVVECLPG.K.DGT..VRVAKLKTKHG.T.............YIRPVTKLC.IF
A0A0N5E799.1/150-243      RKWRAAQAIVESCWKRWQREYVPYLLERRRWVRPTRN.LQV.DDIVLVIDPQNTREHWVLGRVTECNSG.Q.DGT..VRVVRGKTNRG.T.............YLRPVTKLC.LM
A0A085MSJ2.1/1137-1230    RKWRVSQAIVESFWKRWLKCYVPSLIERKKWLKHRPQ.LRE.GDIVVVVDDRIPRGFWPLGRVQECFSG.T.DGV..VRTAKVATKFG.T.............YTRPVTKLC.LL
A0A0N5E596.1/1615-1708    RQWKSAQAIVEHFWARWLREYVPNLIERRKWLRDRRT.LKQ.DDVVLIVDENLPRGHWLTGRVLEVWPG.P.DGL..VRTATVRTARG.D.............FKRPVIKLC.LL
A0A085ND98.1/181-269      KRWKYAQALATHFWRWMKEYVPSLIGRGKWTKNEWN.MSV.GDIVVLVDPNSP.....LGRITTVYYG.K.DGV..VRSADVALALT.A.............TRRSVNKLL.LL
A0A162BZQ0.1/104-197      RRFRQSQFIVNQYWRRWMQEYVPELIERKKWNLVTPP.LRV.GNRVLIMDKNTRCGQWLTGTVSKLFPG.D.DGR..IRRVSVKTATS.E.............LIRPVVKLC.LF
A0A164T6R0.1/947-1043     KQFLQSQAILQHFWNRWFREYVPHLTERRKWTENRPN.VTV.GDLVLVIEPNTLRGQWSLRKVIEVLPSES.DNV..VRVVKVQISNH.K...........NPCIRPVSKLC.VM
A0A085ND67.1/1035-1128    KHWQHAQVIVERFWRRWQRELLPTLQRRTKWMVESPN.LKV.DDVVVIVDPNSPKGQWPLGRIVDVFKG.K.DGR..VRVVNVKTTTG.T.............YKRPITRLT.VL
A0A085NQM3.1/849-942      RQWKIAQTIADRFWKRWLKDYLPTLQTRAKWVVDQPN.LQV.GDFVMVVDPLQKRGSWTTGRVVKVHPG.R.DGI..VRTVTVKTASG.S.............YIRPASRIA.LL
A0A085N844.1/484-577      RHWKTAQLIVNRFWKRWLREYLPTLQRRDKWWNDTAN.LRV.GDFVLVIEPTVKRGEWLTGRVVRVIDS.P.DGI..IRKAIVKTKLG.P.............YERPVTRLA.LL
I3KZ01.1/557-650          RRWKHSQVLADHFWSRFIRLYLPSLQLRPKWRSSPAD.VTE.GSVVMIVDPQLPRASWLVGRISKVHPS.P.DGH..IRSADVKVKDR.T.............FTRPVARLV.VL
M4AYD3.1/1791-1884        RHWRTCQVLSDRFWTQFIRHYLPTLQTRLKWKRDTSP.IQL.GTVVMIVDPQLPRASWPIGEISRVFPG.A.DGL..IRSAEVKVKDH.M.............YTRPVSRLI.RL
A0A085NCZ0.1/915-1008     RSWLLAQAILDRFWRRWMKEYIPTLAPGPNRSADHSS.ITC.GTVVLIVDPNLPRGQWLMGKVTRLFPG.E.DGT..ARVAEVKTKYG.L.............RRRPLVKLL.QL
A0A085ND08.1/853-947      RNWRLTQTILDRFWGRWMKEYVPTLAPTQNKDANQPT.ITR.DSIVLVVDPNAPRGQWLMGKVVHLFPG.K.DGI..PRVAEVQTQYG.L.............KRRPLVKLLELL
A0A0N5DME5.1/207-299      RRWESSQILVNQFWRRWIREYLPSLASPRKGQSGKER.IEK.GDVVLIVETGNPRGVWPMGKVSETYPG.P.DGI..VRVVDVQGSRG.T.............RNPDEPAA.AV
K7JA41.1/943-1036         KQWQIAQFFADSFWRRWLKEFLPTLIPRKKWRDSEDP.LKI.GDLVLILDDSVDRNEWRKGIIIKVFPG.S.DGH..VRVAEVKTAYG.T.............VKRPTRKLV.KF
A0A183VJA5.1/214-301      KRWEEGQHLTDLFWHRWLSEYLPMLQTRSKWLDIKCK.LQP.GDLVLVNS......LWPKATVKQVYYD.Q.SGR..VRTVRLKTATE.E.............VKRNIRSIC.LL
G7YPE3.1/466-559          RQWRQSQLMADTFWRRWLREYLPLLQSRHKWTNLRRN.VRG.GDLVLILGEQHTRGTWSKGVVTATIPG.S.DGC..VRTVVLKTPDG.S.............VTRDIRKIC.ML
A0A183MYV6.1/498-591      RQWRQAQLLATTFWRRWIKEYLPLLQTRTKWTQKRRD.LQV.GDLVLVIGDSYARNNWPKGLVISIDPG.E.DGL..VRQAKIRTSKG.I.............IIRDIRKIC.LL
G7YI41.1/145-238          AGWKQVNYLAGVFWRRWIREYLPTLQVRTKWLRASQG.LKR.GDLVLVSGEDLPREKWPLGIVDSCEAS.P.DGL..VRTLHVRTAGG.V.............LKRDIRKVC.LL
W4ZKP7.1/1717-1809        RRWRQIQYLADLFWRRWVKEYLPSLQQRQKWLKETRN.LEE.GDIVLIID.NLPRNKWLTGRVIETYPG.K.DKL..VRSVKLKHSTG.I.............LVRPVQKVC.LL
W4YNM4.1/144-236          RRWRQIQFLADVFWRRWTREYLPALQQRQKWLVPTRN.LTE.GDVVLIKD.DLPRNQWLMGRIVETFAG.R.DNL..VRTAKVXTKST.T.............LVRPIHKLC.LI
W4ZBS7.1/1652-1744        RRWRQVQYLSDIFWEKWRKENLPTLQARQRWNERERD.VSV.GDIVHIME.EAPRNHWAVGRVVEIFPG.K.DGL..VRSVRVQTKSS.T.............LVRPIHKLC.LL
W4ZDM8.1/1766-1859        KRWRQMQLLADTFWRRWVREYLPTLQQRQKWAQPNRS.LQV.GDVVLIADKDAPRCSWQMGRVEEVYPG.K.DGL..VRSARIKTKST.T.............LTRPVIKLC.SL
```

```
W4XIU6.1/1692-1785      RRWRQMQYLADLFWKRWIREYLPSLQQRQRWQQPERN.LQV.GDVVLIADGTVPRCCWLMGRVLEVYPG.K.RGY..VRSVKVKTATS.T.............LVRPIAKLC.LL
W4ZGF5.1/276-369        RRWRQIKFLADVFWRRWSREYLPQLQRRERWVPSHRN.LQI.GDLVLVADANTPRNSWLLGRVLQTFPD.K.KGF..TRSANIKTKNS.V.............IVRPISKLC.LI
W4YQI7.1/1221-1314      RRWRQVQYMADLFWKRWIREYLPQLQEREKWVASRRN.VQV.GDVVLVVNNNAPRNLWQLGRVLKTMPD.K.GGF..VRSVEVKTKNS.V.............LVRPISKLC.LV
W4Z2K2.1/1636-1729      RRWRQVQYMSDLFWRRWTKEYLPQLQQRQKWVEPHRN.LQV.DDLVLVVDQTIPRNFWLLGRVSRVLPD.A.EGL..VRRAEVKTKLG.V.............YLRPISKLC.FI
W4ZL70.1/1791-1884      RRWRQVQYLADVFWRRWLREYLPQLQARQKWVHPSRN.IQV.GDVVLVVNETAPRNFWLMGVIIGITPD.R.QGM..VRRAEVKTKTG.T.............YSRPISKLC.LL
W4Y4R7.1/1669-1762      RRWRQVQLLADTFWSRWVKEYLPRLQERQKWATSRRN.IQV.GDLVLVVDEHAPRNCWQIGRVLRTMPD.R.EGN..VRQCEVKTQTS.T.............LRRPIAKLC.LL
W4Z131.1/1855-1948      RRWRQVQFLANQFWKRWIKEYLPLLQQRQKWTRVERN.FQT.DDLVLVADDSSPRGKWPLGKVTATYPD.R.HGR..VRQVEVRVGSK.Y.............YKRPISKLC.LL
W4XEJ5.1/1310-1403      RHWRQVQYLAGQFWTRWVKEYLPLLQQRQKWTRPNRN.FEV.DDLVLVVNESTPRGQWPIGRVLTTYPD.K.RGL..VRQVEVRVGPR.Y.............YKRPISKLC.LL
W4Y2R6.1/1821-1914      RRWRQAQYLSSLFWKRWRREYLPLLQERQKWTKPQRN.IAK.DDLVLVVDDNAPRGQWPLGKIVDVFPG.R.DHR..VRQVEVRIGTK.Y.............FRRPITKLC.LL
A0A0N5E5X6.1/215-308    KRWRHSQFLANQFWKRWTKEYLPTLMNRRKWYRVQRN.LKI.GDVVILVEANTPRGKWPLGKIEQVFPS.N.DGA..VRSVLVRSGGT.T.............FHRPVSKLC.LL
G3NZ93.1/640-737        RQWRQVQHLSNTFWDRWRKQYLPTLQARKKWHSAHPN.ISP.GSVVILKDSQAPRNEWPLGLVTQAFPS.E.DGK..VRKVEIKVARTGV.........TKLFLRPVSEIV.LL
V8N7M6.1/1629-1734      QQWRQVQQLANTFWNHWKQEYLSTLQDCHKWKTKELN.LEE.GDLVLLKDKDVHRNLWPMGLVHKILPS.S.DGLVQVKVLIVRL....LDSSCDQQAFTKKLYTRPITDLV.FL
W4ZF56.1/1779-1885      KRWRRVQHLINEFWDRWRKEFLQNLQTRQKWTRPRRD.AKI.GDIVIMKDENLPRNQWSLARISEVFPS.T.DGL..IRKVKLVKGDS.ELDDSGKRKGPVKELERPIHKLV.LL
W4YAI7.1/1012-1117      KRWRRVQHLANEFWKRWKREFLQSLQSRQKWTCPQRE.TRI.GDVVIMKDDN.PRNQWTLARVVDVYPS.S.DNH..VRKVKLVMGDP.KINSRGKRIFQQRYFERPIHKLI.LL
W4YRC9.1/1794-1900      KWWRRVQYLANEFWLRWRREYLQQLQPRQKWTRTQRD.LAI.GDVVIAKETEDVRLQWPLARVAEVFPS.Q.DGR..VRKVKLLMADG.ALDSNGKRLKKPCYLDRPVHKLV.LL
W4YQ55.1/1732-1838      KWWRRVQYAANEFWARWRREYLQELQSRQKWVLKRRN.LEV.GDIVISKENEDARGRWPLARVVDVYPS.K.DGC..VRKVKILIADG.TLDNEGKRKRPPVELDRPIQKLV.LL
W4YL88.1/161-252        KRWRRVQYLTQFFWTRWQKEYLHTLQPRKTWTEKKTN.VHV.DDIVLLKDEAICRTDWRVARVMETMSS.D.DGL..VRKVRLLMATP.D.............LDKKV..FT.AL
W4YNL3.1/1345-1451      KAWRRVQFLLEQFWSRWRKEYLLGLQRRQKWTMPRRN.LQV.GDIVLLTDDEAPRMKWPIAMVAEATPD.E.DGL..VRRVQVRVGTK.DLDNKGRPNKKLAEYWRPVQKLV.LL
K7JBZ5.1/1241-1333      SGYKYKLELMEHLRRRFRKEYLSQLVIKPRITESRK..LKK.GDVVLVGDDSRKRIDWPLARIEELIEG.R.DGK..NRIAILKRSTG.N.............FKRPVQRIY.PL
A0A085MRM7.1/437-527    NRAKYRQRLMKCLRRRFRTEYLGQLLQRPNRGKRPSG.IAV.GDVVLIGSDNQTRLFWPLARVERLIPG.Q.DGE..PRIAMVRTATT.Q.............LMRPLQRL....
A0A087UD40.1/941-1032   LRFKHCQKLRDELRTRFRKEYLGQLVQRAGEKSRS...LKI.GDVVLVGQDNVKRQHWPLARVEKIFPS.R.DGC..PRVAQVKTKIG.I.............LVRPIRKLY.PL
K7JB95.1/1088-1181      RRLKYLQRLRQELRVRFRSEYLGQLKRRAQSNRLFTN.IKS.GDIVLIASDNQKRLDWPLARVIELIPG.N.DGV..TRIARLKTATG.E.............LVRPLQRLV.VL
A0A087SX37.1/313-403    KRLVYRRKLMSDLRVRFRNEYLGHLHQRTIALNRFYV.PNV.GD..LIWTDNLKRLYWPLGRILSIYTS.N.DRV..ARTTKVKTKPG.L.............VTRP.SKDC.VL
A0A087UB64.1/1587-1680  RRVRHRQKLFNDLRHRFRKEYLGLLVQNKIQKGPSSE.LRL.GEIVLIGDDVKKRMHWPLARVVRLIPG.K.DGK..VRTVELKTQAG.V.............LLRPIQRVF.PL
A0A087T6D1.1/283-376    NRYAYQMKVREDLRKRFRTEYLGQLKNHTLKTCLKYVKLSV.GDIVLIETPE.KRIKWPLGKIITLIPG.N.DGI..VRLVKLRTKNG.E.............ILQPIQQIY.PL
A0A087TQK7.1/997-1090   KRVRYLHNLRLNLRKRFYKEYLGELV.RSPKSSSRRKTISP.GEIVLVESKNPNRINWPLAQVIELFPG.K.DNI..ERVAKLRVANG.E.............IIRPLQRIF.PL
A0A087UFP5.1/463-555    KRWQYHQKLCVEFRKRFRDEYLGLLVHRPKTVPLQI..IKE.GDLVFVEDDRRKRTDWPMGCVIETYPR.K.DNN..TRVVKVETSSG.E.............LLRSVQRIF.PI
J9K169.2/1236-1330      RRWKYVQYLMQSFWKRWEKEYLPQCQVRGKWVTNIRP.LEI.DDVVIVKGESTHPTRWKLGRIINLHPG.K.DKV..IRVATVRTANGTE.............MRRPTVKLC.RL
X1WVK5.1/1121-1215      RRWKYVQALMKDFWNRWYKEYLPQLQVRGRWVARKAT.LRV.GDVVIIKEDCTPPSRWKLGIITTVHPG.K.DEV..VRVVTVRTSNGTE.............LKRPVVKLC.RL
X1WWR0.1/1131-1225      RRWKYVQHLSQTFWKRWHAEYLPQMQVRGKWITKKGP.LKI.DDIVIIKEDHVPPTKWKLGRVIKVHPG.V.DGE..IRVVTVRIGSGTE.............MKRPTVKLC.RL
X1XU44.1/1411-1505      RRWKYLQALMQLFWRRWSSEYLPQLQIRGKWTSNKTL.LKI.GDIAIIRDENTPPTKWKLGRVVNVHPG.K.DGI..IRVVTMRLGSGAE.............LKRPTVKLC.LL
J9JVI2.1/212-306        SKWKSVQGLMQTFWKRWHIEYLPQLQVRGKWTAGCKP.LEV.NDLVIVKEDNMPPARWKLARITCVHPG.S.DGQ..IRVVTIRLANGNE.............IKRLVVKLC.RL
J9KBH1.2/1583-1677      KRWKLVQGLLQSFWNRWYAEYVPQLQIRQKWTSGLSP.LSV.NDVVIVKEENLPPTRWKLARVVKVHPG.K.DNV..VRVATVKLANGSD.............LTRPIVKLC.RL
J9LJ20.2/221-315        RRWKLVQALMQSFWTRWTSEYLPQLQVRGKWTTTTKP.LAI.GDVVIVKEDNMAPCMWKMAHITRLHPG.R.DGH..IRVVTIRTANGTE.............TKRPVLKLC.LL
X1XTZ9.1/2741-2835      RRWRFVQFLMQDFWKRWHKEYLPQLQTRGKWTSGTET.FKV.NDIVIVKEENTPPMKWKLARIVQVHPG.T.DGR..TRVVTIHMANGLE.............TRRPVAKLC.RL
X1WT29.1/446-540        RRWRYVQCLMQSFWRRWQTEYLPQLQIRGKWTTNSKE.LAV.DDIVIVKDENMPPAKWKLARIIETHPG.N.DGR..IRVVTIRTANNNE.............MRRPVIKLC.RL
X1WK54.1/930-1024       RRWKFVQHLLQTFWKRWHMEYLPQLQVRGKWTSGSSN.LAI.DDLVIVKEDNLPPFKWHLARVIELHLG.C.DGN..VRVVSVRNSSGKT.............MRRPVAKLC.KL
X1X7H4.1/1590-1684      RRFKLMQAQFQIFWKRWLSEYLPQCQRKGKWLKRTRS.AVV.GDIAILKNELLPPLQWPLVRITEIHPG.R.DGI..VRAVTVRNSAGQL.............FSRPVVKLA.FL
J9KV52.2/2624-2718      RRFKLMQSRVQVFWKTWSTEYLPQVQRRGKWTKMCRN.VVI.GDLAILKNENLPPMQWPLVRIIKVHPG.M.DNV..VRAVTVRNSSGLE.............FKRPVVKLA.II
J9L9F8.2/999-1093       KRYSLMRQQMQLFRKRWSQEYLPQLHRRGKWTKPNRN.FCV.GDLAVLRDDNMPPLKWPLVRVAAVHPG.A.DGV..VRAVTVRNSTGSE.............FRRPAIKLS.LL
J9JWX5.2/1608-1702      KRFALVQSCIQQFWKRWSREYLPQLQRRGRWIKVTRN.MRV.GDLCLLRQENLPPTRWALVRVQDVHPG.P.DGT..VRVVTLRNSSGNT.............FKRPVVKLS.LL
J9LMD2.2/253-348        RKFQLIQARLQHFWKRWSAEYLPQLQRRDRWISKKSKDISV.GDLAILKPDNIPSLQWQMVRVIKVCPG.A.DGV..VRVVTVRNATGTE.............YVRPVRKLA.RL
X1WV59.1/1259-1353      KRWELVKLQSQTFWKRWSTEYLPQLHKRGKWLSKKDT.IRV.GSLAILREDNLPPLQWKMVRITKVHPG.P.DGI..IRVVTVTNSAGRE.............FQRPAVKIA.VL
X1XMP6.1/225-319        KRLELVRVQAQTFWKRWSSEYLPQLHKRGCWLSKKDN.IEV.GSLAILKEDNLPPMKWIMVRVTQVHPG.S.DGM..VRVVTVRNSAGRE.............FQRPTAKLA.VL
A0A1X7TFI7.1/389-490    KRWQLCQTLVSHFWKRWSSEYISTLNRSNKWTSNRRN.LTT.GDIVLLHEDNVIPTQWPLAKVIQTHPG.E.DGL...NYTYFTTDQG...LGGQGKMPHLPPYWQPCKQ...AI
A0A1X7UAA5.1/1713-1806  KRWHLCQSVLNHFWKRWSGEYISSLQRYAKWNTSTRN.IEV.DDVVIVREDGLLPTKWAMARVIGVHPG.Q.DGI..VRVVTIKTPNG.T.............YKRPVTKVA.VL
X1WL73.1/1147-1241      DRWRRVQQFSQILWKRWSREYLHQLQERSKWASEKGPKVDV.GSVVLISEDNLPPLRWKIGRVREVTRG.N.DNI..IRTAVVKTADG.E.............LTRAVRRLC.PL
J9LVB4.2/1622-1716      TRWRRLTHYSQIIWKKWSREYLNQLQERKRWAGEKGPRIDI.GTVVLVRDDNISPLGWKLGLVTNIQRG.T.DNV..IRSAEVRVGKG.C.............FTRSVRNLC.PL
X1WNQ0.1/1107-1201      DRWRRVRQYTQILWKRWSNEYLNQLQVRKKWSVERGPTLAI.GTIVLMRDENIKPLNWKLGRVLSVSHG.E.DGV..IRTALVKTMTG.E.............YNRAVRNLC.PL
X1XH24.1/1220-1314      DRWRRVTKYSQTLWNRWSTEYLNQLQERVKWAGAKGPSVRI.GTMVILRDTNLPPLQWRLGRVVKVEPG.A.DGV..IRAASVQTSSG.V.............WKRAVRLLC.PL
J9KLD6.2/1315-1409      TRWRRVTQCSQLLWRRWSTEYLGQLQERAKWSSEKGPGIRL.GSVVLIKEDNIPALQWRLGVIVSTPQG.D.DGI..IRVAQVKTAEG.T.............YKRAVRQLC.PL
```

```
J9LHB7.2/1632-1726      NRWRRVTQAFQQIWSRWQKEYLAQLQQRGKWESEKGPRVKI.GTVVLMKEDNIPPLQWKLGKVVQLHTG.P.DKV..VRVVTLLSAKG.Q.............FTRAIRMVC.PL
J9KBV5.2/1408-1502      KRWQRVTQLTQTLWQRWNKEYLSQLQTRRKWLDNKGPKLKV.GTVVLVKENDTLPMSWSMGRVVRVQAG.D.DGV..IRVATILVKGK.E.............VSRAVRKLC.PL
A0A0J7NFF3.1/354-448    LRWQRVEQMRQHFWKRWNSEYLHTLIERQKWRINKGEQLKV.GRLVLIQQSGLGPLQWLLGRVLQVHLG.V.DGI..ARTATLRTSKG.C.............LTRPLTRLA.IL
A0A0J7K7S0.1/830-924    SRWQHVEQMRQHFWNRWSTEYLNQLQQRTKWKVSGGRQLLP.GCLVLVQQRGLTPLQWILGRVIKVHPG.A.DGI..ARTATIRTSKG.V.............LDRPLTKLC.IL
A0A026W4M6.1/139-233    TRWQRVEQIRQHFWRRWSAEYLHSLQQRSKWKTNKGVQLQP.NQMVLIKQQGLAPLQWLLGRVEEVHPG.S.DGV..ARSAAVRTAKG.L.............IVRPLFKLA.IL
A0A151XBN9.1/410-504    LRWQRVSQIFQHFWKRWSREYLQQLQQRTKWQRSKGPHPVI.GDMVMVQENNSLPLQWAVGRIEDVYSG.T.DGV..SRVVSVRTVKG.T.............YKRPITKIC.IL
A0A0J7K277.1/104-197    SRWQYIEQLRQHFWKRWSLEYLHHCQQRNKWQTQNVS.IHV.GQMVLLKEDNIPPLSWPLGRIQEVHPG.K.DDI..IRIATVRTNKG.T.............YKRSLTRLC.LL
E9J9Y8.1/1146-1239      SRWQYVQRLRRNFWQRWTQEFLHLCQQRNKWNIENPEVLP..GQMVLIKDDTAPPLSWALGRVQETHPG.T.DGI..VRVATIRTANG.T.............YKRPITRLC.LL
A0A0J7KFN2.1/625-717    SRWQSVEQLKQHFWHRWLKEYLHNCQARVKWNTTNDP.IKV.GQMVILQED.LPPLCWSLARVEEIFPS.K.DNI..IRVVSVRTPKG.I.............YKRPITKLC.IL
X1X9I2.1/525-619        SRWKKVQNLVQQIWRRWSVEYLSQLQERKKWDKSRGPSVKV.GSMVIVRDTNLPPLQWHLGRVIDVFPG.K.DGV..VRVAMINTASG.P.............KKRAVRLLC.PL
A0A0J7N453.1/121-214    STWQHISKVRQDFWTRWNLEYLNELQMRNKWHKDGAK.LEV.GTVVLIKEKNLPCTQWAMGRIKEVHPG.G.DGV..IRAATIKTATN.E.............IKRAAKMLC.PL
K7JP64.1/220-313        SIWQFICKARQDFWKRWHIDYLSELQKRQKWFEGKGE.ITP.GSIVIIIDKNQQCNQWPLGKVLEIYPG.K.DGI..IRVAKIRTKTG.E.............YIRNVILLC.PL
D7EKM8.1/1234-1327      TRKELLDQILQTYWKRWHVEYLHNLQVRQKWNKPSSP.IKA.GTVVVLRTDNTPPLHWPLGVVQEVFPG.K.DGI..VRVASVKTPNG.L.............YKRPIVRLC.PL
A0A0L7QJG1.1/222-315    SSWQQIQKVKQHFWARWHREYLNELVTRSKWSSGSHT.ITE.GTIVLLREDNIPPMQWALGRVTQVHPG.S.DGI..VRTVTIKTATN.V.............LVRSVKKLA.PL
A0A0L7QJB6.1/116-210    STWQHVQKMKQHFWTRWHKEYLHELTVRRKWHRGQTNDLPV.NTMVILHEDNAPPIRWPLGRITEVHPG.E.DGV..IRVATVRTTNG.T.............YKRSIKKLS.PL
A0A0J7N6J8.1/169-263    STWQHIQKIKQHFWNRWNKEYLNELQQRTKWLPSKPHGIGV.GDFVILKEDNTPPLHWITGRVIVTHPG.D.DGV..VRVVTVKTVSG.T.............YKRCIKKVS.PL
B4K1B0.1/373-466        ERFDVITAAKQQFWRRWSSDYIHELRARTKWTSSSSN.LAI.GTMVIIHDDNLPAQQWKLGRIEALVPG.K.DGH..VRVVHLRTANG.I.............CWRPVHKLA.TL
B4HCP3.1/239-333        KRWRLVSSARQMFWQRWSREYVLGLQIRCKWHQEEPN.IKE.GDLVIVAEDNLPPQHWLLGRVVGTTAG.Q.DGR..VRVVDLRTSSGAT.............FRRPIHKLA.LL
B4HCL7.1/368-461        NHWKRILMVHHMFWHRWSSEYLTLLQKRTKWNTVANN.IQL.GTLVLIAEDNAPPGQWLLGRVAELHPG.T.DGA..VRIVTLRTKTG.L.............FKRNVHKLC.PL
A0A0J7KH68.1/87-180     DRWQMIQRTFQDFWKRWAAEYLNNLQGRSKWQTAREN.LQI.NDLVIVREENIPPLGWKLGRVIELHPG.D.DGR..VQVAIVRTSGG.N.............IKRSIAKLC.KL
A0A087UCD6.1/590-678    DRWKNLQRLFQQFWKRWSSEYLSRLQQRPKWCKLQRD.LKI.GDMVLIKNENLPPLRWKLGRIVKVYPG.L.DDR..IRVVDLRTSSA.N.................NNLT.PL
```